An Order Processing System for a Corporate Clothing Company

Adam Raymond
Computing
Session 2004 / 2005

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student)_____

Summary

The overall aim of the project was to design and implement a web-based order processing system for Laurence Highman & Uniformity. The London based company, specialise in the manufacture and distribution of corporate clothing garments for a wide variety of customers including those in security, roadwork maintenance and high street stores market sectors.

The nature of the company's business has changed in recent years, resulting in an increase in the number of garments imported from overseas. Laurence Highman & Uniformity therefore recognised the need for a system that would aid them in managing the processing of imported garments. The proposed system would have to emulate the paper system as closely as possible with some further enhancements and therefore, improve the overall efficiency of the ordering process.

The project required the developer to adhere to project management techniques in order to correctly and successfully gather the company's requirements, prior to designing, implementing, testing and evaluating the system.

Keywords:

Java, SQL, MySQL, Tomcat, JDBC, JSTL, JSP, HTML, CSS, STRIDE, Security, SSL, Model View Controller, Geary, Goods, Clothing, Order-lines, Logging, Auditing, SDLC.

Acknowledgements

Firstly, I would like to thank my project supervisor, Dr. Kristina Vuskovic and the project coordinator, Dr. Sarah Fores, for their help and guidance throughout the project on various matters.

I would also like to thank Mr. Owen Johnson, for his valuable feedback during the progress meeting in March 2005.

I would like to thank my personal tutor, Dr. Kevin McEvoy for his help and support over the past three years.

I would like to thank Alan Gabbay, Jonathan Vickers and David Boda for their valued friendship throughout the project and over the past three years.

Finally, I would like to acknowledge Mr. Michael Rothschild, and all the staff at Laurence Highman & Uniformity for dedicating their time in interviews and the software evaluation.

Project Table of Contents

Summary ..... **i**
Acknowledgements ..... **ii**
Project Table of Contents ..... **iii**

1 Introduction **1**
1.1 Background to Laurence Highman & Uniformity 1
1.2 Problem Recognition 1
1.3 Project Aim and Objectives 1
1.4 Minimum Requirements 2
1.5 Deliverables 2
1.6 Project Relevance 2
1.7 Project Management 2
1.8 Report Outline 2

2 Background Research **3**
2.1 Methodology Selection 3
2.1.1 System Development Life Cycle 3
2.1.2 The Spiral Model 4
2.1.3 The V Model 4
2.1.4 Final Methodology Choice 4
2.2 Database System Selection 5
2.2.1 Microsoft Access 5
2.2.2 PostgreSQL 6
2.2.3 MS SQL Server 6
2.2.4 MySQL 6
2.3 Web Server Selection 7
2.3.1 Microsoft Internet Information Services (IIS) 7
2.3.2 Apache Tomcat 7
2.4 System Architecture Selection 8
2.5 Development Language Selection 8
2.6 Scripting Language Selection 9
2.6.1 Practical Extraction and Report Language 9
8.6.2 Active Server Pages 9
8.6.3 Hypertext Pre-Processor 9
8.6.4 Java Server Pages 10
2.7 Security 10
2.7.1 Secure Sockets Layer Capabilities 10
2.7.2 STRIDE Analysis 11
2.8 Usability Principles 11
2.9 Project Amendments 12

3 Analysis 12
3.1 Current Systems Summary 12
3.2 Current Business Summary 13
3.3 Analysis of Current Business Processes 14
3.3.1 Current System Shortfalls 15
3.4 Feasibility Study 16
3.4.1 Technical Feasibility 16
3.4.2 Economical Feasibility 16
Cost Analysis 16
Benefits Analysis 17
3.4.3 Legal Feasibility 18
3.4.4 Operational Feasibility 18
3.4.5 Schedule Feasibility 19
3.5 Methods of Gathering Information 19
3.6 Requirements List 20
Data and Information Requirements 20
3.6.2 Functional requirements 20
System Features 20
Constraints 21
3.6.3 Non-functional Requirements 21
Usability 21
Security 21
Performance 22
Maintenance 22

4 Design 22
Model View Controller Benefits 22
4.1 Model 23
4.1.1 Java Database Connectivity 23
4.1.2 Prepared Statement Execution 23
4.1.3 Database design 24
Table Normalisation 24
4.2 View 24
4.2.1 Java Server Pages Design 24
4.2.2 Java Server Pages Design 25
4.3 Controller 25
4.4 User Interface 26
4.4.1 Usability Design 26
4.4.2 System Navigation 27
4.4.3 JSTL Design 28
4.5 Component and Deployment Design 28

5 Implementation 28
5.1 Database Implementation 29
5.2.1 Tables 29
Primary Key Applications 29

Data Types 29
Other Imposed Constraints 30
5.2 Server Implementation 31
Enhancements to the Tomcat Configuration 31
5.3 Client Application Implementation 31
5.3.1 Laurence Highman & Uniformity Framework 32
Categorising Action Types 32
Other Validation Types 32
Constants Implementation 33
5.3.2 Java Database Connectivity 33
5.3.3 Report Generation 34
5.3.4 Item Check-In Monitoring 34
5.3.5 Report Search Facilities 35
5.3.6 Persisting Updated Information 35
5.3.7 Retrieving Data for Display 35
5.4 User Interface Implementation 36
HTML Layout 36
JSP Implementation 36
JavaScript Implementation 37
CSS Implementation 37
5.6 Security Implementation 37

6 System Testing 38
6.1 Sample Data Deployment 38
6.2 Unit Testing 38
6.3 Black Box Testing 39
6.4 White Box Testing 39
6.5 Integration Testing 39
6.6 User Acceptance Testing 40

7 Evaluation 41
7.1 Meeting the Minimum Requirements 41
7.2 Exceeding the Minimum Requirements 42
7.3 Potential System Enhancements 42
7.3.1 Determining Carriage Costs 42
7.3.2 Improved Administrator Features 43
7.3.3 Generating Order Quotations 43
7.3.4 Customisable Interface and Page Layout 43
7.4 Evaluation of Chosen Technologies 43
7.5 Evaluation of Chosen Methodology 44
7.6 Evaluation of Implementation 44
7.7 User Evaluation 45
7.8 System Criticism 46
7.8.1 Improved Concurrency Controls 46
7.8.2 Logging and Auditing Enhancements 46
7.8.3 JavaScript Enhancements 47

Bibliography **47**

Appendix A – Project Reflection **50**
Appendix B- Project Schedule (Original Gantt chart) **51**
Appendix C - Project Schedule (Revised Gantt chart) **52**
Appendix D – Current Documentation **53**
Appendix D1 – Sample Purchase order for 'Rael Brook' 53
Appendix D2 – Sample order book excerpt for 'Rael Brook' 53
Appendix D3 – Sales Invoice from 'Rael Brook' 54
Appendix D4 – Sample Purchase order for 'Regatta' 55
Appendix D5 – Sample order book excerpt for 'Regatta' 56
Appendix D6 – Sales Invoice from 'Regatta' 57
Appendix E – Use Case Diagram **59**
Appendix F – Activity Diagram **59**
Appendix G – Package Structure of System **60**
Appendix H – E-R Diagram for Proposed Solution **61**
Appendix I – System Concept Level Class Diagram **63**
Appendix J – Navigation Map for Proposed System **64**
Appendix K - Database Schema **65**
Appendix L – System Screen Shots **66**
Appendix M – Unit Testing Results **69**
Appendix N – Black Box Testing Results **86**
Appendix O – White Box Testing Results **89**
Appendix P – User Acceptance Testing **92**
Appendix Q – User Evaluation results **95**
Appendix R – Purchase Order – Implemented System **96**
Appendix S – Letter of Thanks from Laurence Highman & Uniformity **97**

1 Introduction

The purpose of this chapter is to provide company background information and the overall aims for the project. This chapter also outlines the format for the rest of the project report, and briefly discusses items such as the deliverables and problem relevance to the developer's degree programme.

1.1 Background to Laurence Highman & Uniformity

Laurence Highman & Uniformity is a London based corporate clothing company. The company customises and supplies uniforms and other corporate clothing for a wide range of customers to include airlines, security and banquet companies. Up to 1985 the company was run by Mr. David Rothschild, where the business was in need of technology upgrades and began to make losses. Mr. Michael Rothschild took charge in late 1985 and helped turnaround the business with modern IT usage and state of the art cutting machinery at the time. The business began to recover within a few years and gradually an increasing amount was being successfully reinvested into the company.

In recent years the business has been facing a number of transitions. Particularly in the past five years, many competitors in the industry have been importing garments from overseas. This has meant that in order for Laurence Highman & Uniformity to survive they have been forced to follow suit. The company now imports nearly 70% of all garments from overseas. These garments are then tailored to the customer's requirements.

1.2 Problem Recognition

The business transitions have meant that the company do not currently have a system that can handle the processing of orders sufficiently. The company recognise that their paper based system is prone to errors, and mobile sales representatives need to contact the company via telephone for information, which disrupts in-house tasks. The time that could be spent increasing order intakes and promoting order-lines, seem to be spent on resolving human errors and communication issues with mobile sales persons.

1.3 Project Aim and Objectives

The overall aim of the project was to gather information regarding the current system and formulate a solution. Following this, design and implement a web-based order processing system that makes daily operations for orders more time and error-efficient for Laurence Highman & Uniformity. The overall objectives are to use appropriate tools to design and implement the system, in addition to an appropriate methodology selection. The new system should also mirror the existing paper based system with additional features.

1.4 Minimum Requirements

The minimum requirements aim to construct dependable foundations, in order to cater for extensions that can be added later. Therefore, the minimum requirements must be able to solve the problem. The following requirements are considered fundamental for meeting the aims and objectives:

- Summarise the current corporate clothing systems and business.
- Perform a feasibility study of the proposed solution.
- Produce a database solution incorporating web-based access.
- Compose a User Manual to compliment the solution.
- Perform a basic evaluation of the new system.

1.5 Deliverables

In order to achieve the requirements and extensions, a set of goals have been devised. The following list defines the deliverables for the project:

- A Web-based order processing system for Laurence Highman & Uniformity.
- User and Administration Operating Manuals
- Javadoc Documentation
- The Project Report

1.6 Project Relevance

The project draws skills gained from a wide range of taught modules. Information Systems (IN11) and Human-Computer Interaction (SI13) allowed the application of appropriate methodologies and usability consideration. OO Programming (SE21) and OO Analysis and design (IS21) applied concepts learnt for solution implementation and design modelling respectively. Advanced Databases (DB31) and Distributed Systems (SY33) helped define sound database design and the web-based solution respectively.

1.7 Project Management

The revised Gantt chart can be found in Appendix C which illustrates the tasks undertaken in the time period. More time was allocated in the second semester to compensate for less taught modules during this period. Such a project requires the tasks to be dissected into smaller phases. The methodology chosen (Section 2.1) is reflected in the revised Gantt chart for an appropriate course of events in the project time.

1.8 Report Outline

The project report follows an acute variation of the Systems Development Life Cycle. Chapter two details the technologies and methodology chosen, in addition to the reasons they were chosen for. Chapter three

conducts a feasibility study into the proposed system, in addition to forming requirements for the proposed system based on current systems and processes analysis. Chapter four defines the design detail concerning database, framework and front-end interface based on the requirements gathered in chapter three. Chapter five details the implementation for the presentation, application and data layers of the solution. Chapter six describes the testing strategies imposed on the solution and the outcomes of those tests performed, in order to determine if those requirements stated in chapter three have been met. Chapter seven evaluates the finished solution, tools chosen and methodology effectiveness.

2 Background Research

This chapter compares a selection of methodologies, their relevance to good project management and justification of the chosen methodology that the project will adhere to. Following this tools are evaluated and selected for the development of the client's proposed system. This chapter also touches on issues of security and usability and their relevance to the system's performance.

2.1 Methodology Selection

According to Avison and Fitzgerald [1], a methodology in general terms is following the course of a recommended series of steps and procedures for developing an information system. Methodologies help define processes to be broken into phases and thus, what tasks are required to be performed at each phase. A range of methodological approaches exists and are explained by Avgerou & CornFord [2]. An investigation as to which methodologies are most appropriate to the project and client's requirements will therefore take place.

2.1.1 System Development Life Cycle

The System Development Life Cycle (SDLC) is one of the more traditional approaches to software project management. As implied by Avgerou and Cornford [2] the concept is to advance the project systematically through each of the phases with a standard set of outputs at each phase. Although Hughes and Cottrell [3] criticise this particular model for lack of iteration, the methodology can be useful when the production of deliverables are an end result. Hughes and Cottrell [3] also explains that if a particular phase needs to be refined, the traditional SDLC methodology is unaccommodating.

Furthermore, as time is consumed defining requirements, the methodology jeopardises the project as producing a solution could take longer than expected. The company has a firm idea of what the solution should perform and the fact that Laurence Highman & Uniformity wants the solution to emulate the paper system, the requirements would seem fairly set in stone. However, depending on the scale of the project, a

pure linear SDLC approach might create a situation where the solution exhausts time allocated and goes over budget.

2.1.2 The Spiral Model

Avison and Fitzgerald [1] explain the spiral model as a means to help mitigate risks. The model implies the use of the waterfall model for each step. Only the most important requirements are defined and implemented first. After feedback with the client more sophisticated functionality is implemented. However an immediate problem with this approach is that if the client has unintentionally missed an important requirement late into the development process, such changes could prove very costly. This can occur easily if client, developer or even both parties misunderstand the requirements specification.

Avison & Fitzgerald [1] however, do explain that providing 'risk assessment and reduction' occurs during the project, reduces the uncertainty concerning the overall outcome of the solution. However this would rely heavily on the assumption that all staff are knowledgeable enough to alert developers of potential risks. In addition this methodology is said to be more geared to 'large scale projects' and therefore is unlikely to be an appropriate course of actions with Laurence Highman & Uniformity.

2.1.3 The V Model

The V-Model aims to confront the limitations imposed by the traditional SDLC approach and the more accommodating spiral model. Referred to as the 'V-Process Model' by Hughes & Cottrell [3] this model adheres to an SDLC approach until such time that the Implementation phase has been reached. At this point the methodology shifts to become flexible, allowing developers to follow an iterative approach by revisiting the subsequent phases. However this approach may not be suitable to the project as it requires an excessive amount of time to revisit the requirements. Given the limited time span of the project this approach would not be ideal.

2.1.4 Final Methodology Choice

It has been decided to adopt an approach of the SDLC while incorporating an iterative approach over the implementation and testing phases. As mentioned above the client remains firm about what the system must and must not do, and therefore excessively revisiting analysis and design phases of the project consumes valuable project time. Furthermore, in order to ensure that the implementation provides a working solution for the client, an iterative approach for those phases will occur as depicted in figure 2.1.4. This ensures that the software is completed within the given time frame and ample time is provided for potential user evaluations and user acceptance testing, labelled in Figure 2.1.4 as 'Support'.

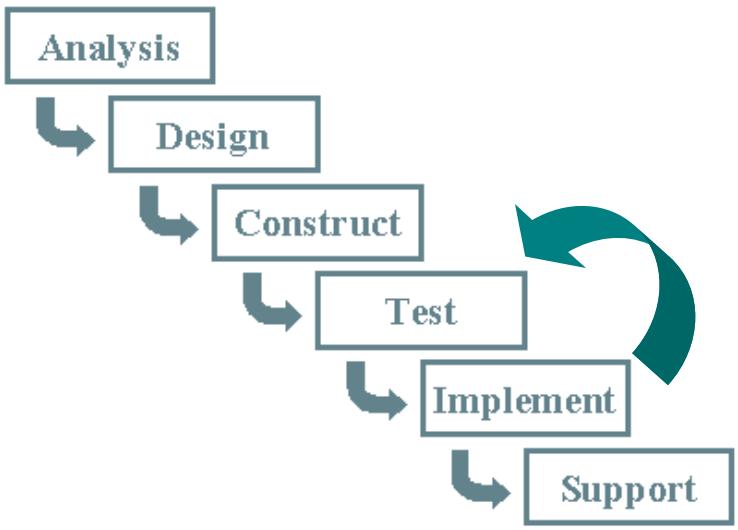


Figure 2.1.4 – Adapted from measureit.com [4]

2.2 Database System Selection

A database is defined by Elmasri and Navathe [5], as a collection of related data. A relational database system would be needed in order to successfully model the relationships and entities identified for the client. In addition, a database system would resolve the inconsistency and information integrity issues and reduce the need to unnecessarily duplicate data. A number of databases are available to use for the application. However, the database systems most appropriate to the problem are MS Access, PostgreSQL, MS SQL Server and MySQL.

2.2.1 Microsoft Access

Microsoft Access is a database system, normally shipped along with the Microsoft Office Suite. The company at present have Microsoft Access available on all their Windows based machines. If using this database system, the company would not need to endure any further costs. In addition, user familiarity of this system is likely to be high at the company. In addition, the fast turnaround of applications using this application would inevitably mean that the implementation phase of the project is quicker. However, Access has the disadvantage of failing to manage multiple accesses to its database efficiently, without causing lockouts temporarily, as explained by igrep [6]. This may prove problematic with constant accesses for example, if all users in the company are heavily using the proposed system. However, this might present more of a problem when sales persons are accessing the system away from the office, as they will require access whenever possible. Finally, in-house support would want to look after the server with remote access but only a number of scripting languages would support the use of Microsoft Access, as explained by db-review.com [7]. Although using Microsoft Access would not cost the company anything extra, it has been ruled out for lack of flexibility when considering concurrent access and scripting language choice range.

2.2.2 PostgreSQL

From background research at db-review.com [7], it would appear that PostgreSQL is an immediate choice over its closest rival, MySQL. PostgreSQL seems to overcome all the shortfalls of MySQL. For example, it is not currently possible for MySQL to create support views or trigger management, as explained by Wiley [8]. However, one downside of PostgreSQL is the fact that it performs considerably slower within a web-based development environment, according to [7]. Taking into consideration that one of the requirements is that users who are not in the office will be able to access the system as seamlessly as possible; the use of a slowly responding database would not be favourable in this particular situation. Furthermore, as the entire system is to be deployed as a web-based application response times of a database becomes a requirement of a higher priority than most. For these reasons, PostgreSQL has been ruled out as a possibility of being used.

2.2.3 MS SQL Server

SQL Server has been developed by Microsoft Corporation. It is considered to be extremely powerful, in addition to offering features such as trigger management, as explained by Harrington [9]. However, such software packages are excessively expensive to purchase, at least for small businesses, as implied by Microsoft themselves[10], and therefore considering the company wish to keep expenses low, is not deemed appropriate for the solution. In addition, SQL Server is limited solely to the use off Microsoft's own Internet Information Server (IIS) which means that the company would not have the flexibility to change their web server should their requirements change. Furthermore, although not an issue at present as the company uses Microsoft Windows as their primary platform, choosing SQL Server may prove problematic should the company not wish to be tied down to a particular Operating System Platform in the future.

2.2.4 MySQL

MySQL was originally developed as a database for open source platforms, and is commonly used with Linux servers. According to Sklar [11], MySQL has the benefit of very small amount of unnecessary coding. The rapid increase in connectivity between MySQL and PHP (discussed later), will have to be taken into consideration, when also determining Server-Side Scripting. Some disadvantages of such an implementation include the fact that MySQL is not able to implement triggers and be efficiently supportive, where the database can have many 'front-ends', as suggested by Wiley [8]. Although MySQL does possess some shortfalls as shown when compared to PostgreSQL earlier, it has proven itself in database robustness and reliability according to db-review.com [7]. In addition, MySQL is entirely platform independent and open source. This means flexibility and cost savings for the customer. In

addition, the company's in-house support staff will be able to remotely support the database on a web-hosting company's set of machines. Db-review also [7] favours MySQL for its excellent scalability, which implies that the client will be running a database that will still function efficiently as the order intake increases. For such reasons, MySQL has been chosen for the database server of the system.

2.3 Web Server Selection

According to Tanenbaum [12], a web server is defined as a server that supplies the requested web pages by the user to their client software. The company plan on hosting the system over the Internet and therefore, the choices made regarding a web server may tie the company down to a narrower range of ISPs. Microsoft's Internet Information Services (IIS) and Apache Tomcat will be considered for the company.

2.3.1 Microsoft Internet Information Services (IIS)

Internet Information Services (IIS) is a web server application which is incorporated into some versions of the Microsoft Windows Operating System, Microsoft [13]. From a support point of view the system is easy to install and configure. Microsoft aim for this software to be user friendly and as expected allows integration with other Microsoft products. However, as much as Microsoft boast benefits of considerably unparallel reliability and security strengths, they have also been releasing a number of security patches and enhancements from time to time regarding this product.

From an administration perspective, the in-house support staff at Laurence Highman & Uniformity will be spending unnecessary amounts of time fixing security and maintenance issues. Furthermore, considering that the proposed system will be deployed over the internet it would not be advisable to favour a web server posing such security vulnerabilities. It is bound to the Windows operating system which although not an issue for the company at present, may be in the future. For such reason it has therefore not been considered as a web server for the implementation.

2.3.2 Apache Tomcat

Apache Tomcat has the advantage of being open source and therefore, would be able to run on both a Windows and Linux platform. According to Wiley [8], this web server's popularity stems from renowned scalability and configurability, which makes it a very popular web server implementation for a variety of requirements. Furthermore, Apache Tomcat does not require much maintenance once installed and it is relatively simple to enhance the configuration and to include features such as SSL support as shown by Jakarta [14]. From a support perspective, the company's in-house support staff will more likely have an

easier task in administering an Apache Tomcat Web Server. For this reason and good reliability it is the web server of choice for the system's implementation.

2.4 System Architecture Selection

As the system will be deployed over the Internet, some form of client-server architecture would match the characteristics of the planned system most appropriately. Tanembaum [12] identifies three layers at which different components are split at different ratios between the client and server side.

The presentation layer is concerned with the display of information on the client's machine. The data layer normally looks after the physical records that are stored in the database sitting on this layer. This layer is normally deployed on the server side, as a method to ensure data integrity and consistency. The application layer somewhat sits between the data and presentation layers. This layer is responsible for gathering users information inserted at the presentation layer to populate the data store at the data layer. The reverse is true when information is being retrieved. For example data will go to the presentation layer after being fetched from the data layer but business rules as defined in the requirements will define what operations are performed before the data gets to the view.

Considering that sales persons will require to access information when out of the office, the system will represent some form of 3-tier client architecture as described immediately above. Such an architecture where the processing occurs all at the server side will maintain data integrity. In addition, by placing all processing logic server side, the system will ensure that no extra components would be required to be installed on the client machine, wherever the client machine happens to be. Three-tier architecture will also cater for the flexibility and scalability requirements that the company have foreseen.

2.5 Development Language Selection

Java has been chosen as the development language for the 'application' layer as described immediately above. This Java implementation goes hand in hand with the use of Java Server pages (see section 2.6). As explained in section 2.6 JSP allows the presentation layer to be rendered without affecting other layers. Geary [15], explains the use of framework known as the 'Model View Controller' pattern. This would allow the separation of content generation from content presentation within the proposed system, enabling a more straightforward implementation phase.

2.6 Scripting Language Selection

Scripting Languages are such that they are interpreted at run time rather than compiled beforehand, as with traditional programming languages. Scripting languages can manipulate data, as well as influence how the user sees the information.

2.6.1 Practical Extraction and Report Language

Practical Extraction and Report Language (PERL), is an open source scripting language. As it is open source, it has the advantage to be available to a variety of platforms, in addition to being reputably reliable, according to Quigley [16]. PERL's strengths lie in comprehensive database connectivity. Unfortunately, PERL is also renowned for being a language with awkward syntax and therefore can be very difficult to both learn and understand. In addition, HTML code cannot be embedded into PERL coding. For such reasons, it was deemed unsuitable for the project. The web based system will be required to consequently display web pages. PERL does not be feasible to implement, due to time constraints and its limitations in terms of operability with HTML syntax.

8.6.2 Active Server Pages

Active Server Pages (ASP) is another development from the Microsoft Corporation. It too, is not platform independent and therefore, ties users down to the Windows operating system, according to Gray [17], who argues that ASP's syntax is not easy to comprehend and fails to handle text based information as well as some of its competitors. Another shortcoming is that this scripting language would not perform without the implementation of IIS, which has already been ruled out by the findings. Therefore, such an approach is unlikely to be successful, given the time constraints of the project. Although the limitations to the Windows operating system are not significant, the potential problems of using ASP may become more apparent once sales persons attempt to use the system away from the office. In conclusion, ASP would not be suitable due to its limitation for sole use with IIS, the Windows operating system, and cost of purchasing the software.

8.6.3 Hypertext Pre-Processor

Hypertext Pre-Processor (PHP) has a distinct immediate advantage, created primarily for internet software deployments. PHP is also considered to enhance rapid product development, due to its subtle learning curve in comparison with other competing scripting technologies, according to Gutmans [18]. PHP also has the advantage of being open sourced and comes with a hefty library, increasing its application power. PHP has also been linked with MySQL, allowing more productive connectivity between the two mediums, according to Greenspan and Bulger, [19]. PHPs rapid application turnaround

and easy learning curve, as well as requirement relevance might be suited to this project. Finally, PHP and MySQL would perform together as an effective implementation according to db-review, [7]. However, while PHP has been mentioned for its rapid application turnaround, Gutmans [18] implies that as there is no distinction between the three tiers of a client-server architecture represented within a PHP page. This means that the more complex the requirements and larger the system the more difficult the pages are to construct and maintain. It was therefore decided that in order to ensure that the requirements have been met, the system should consider the deployment of a server-side scripting language which can differentiate itself from the rest of the three tier client-server architecture.

8.6.4 Java Server Pages

Geary [15], explains how Java Server Pages (JSPs) are a mixture of embedded HTML within Java code on a page. This results in a more dynamic output of web pages that will adhere to an HTML layout. The benefit over PHP with this technology is that the page purely concerns the presentation layer of the 3-tier client-server architecture. Furthermore, this makes the page coding procedure more straightforward and easier to understand. One advantage of JSP here like with PHP, is that no extra features are required at the client-side as the JSP is executed server-side, as explained by Bergsten [20]. Furthermore, Bergsten explains that dynamic presentation is enhanced by the use of JSP Standard Tag Library (JSTL). This allows for the simple formatting of data within the JSP page. For example, one can specify the currency format for a particular field returned from a database, without having any effect on the database itself. Alternatively to JSTL, scriptlet code could be used but as Geary [15], suggests JSTL is far neater in appearance and easier to implement. For such reasons, including a clear distinction of JSP from the other architectural layers, JSP is the chosen server side scripting language.

2.7 Security

2.7.1 Secure Sockets Layer Capabilities

Secure Sockets Layer (SSL) as explained by Whyte [21] is a means by which a client and server can identify themselves over a secure connection. This is performed by the use of sharing public keys using the Public Key Infrastructure (PKI) Mechanism. Once the client and server have identified themselves, the two parties can both transmit and receive data for the duration of the SSL session. Howard et al [22] implies that one should use at least '128-bit cipher strength'. In terms of compatibility for the company, most common Web Browsers are SSL enabled, such as Microsoft Internet Explorer [22].

Considering the system will potentially be deployed over the Internet, sensitive company data may become vulnerable between the transmission of client and server. Whyte [21] discusses a number of

problems with leaving sensitive data unprotected. Two common problems Whyte mentions are impersonation and spoofing. The former concerns a person pretending to be someone they are not. The latter concerns a person managing to intercept the transmission between the client and server for personal gain. In the case of the company, sensitive order information could potentially be snooped upon if unprotected. Web Hosting companies do offer SSL support such as Nameonthe.net [23]. Some ISPs can provide a certificate signed by a trusted third party. Alternatively, one can be created on the command line as Jakarta Documentation explains [14]. The downside of creating one without a trusted third party is that it will be hard to gain trust in the certificate, Whyte [21]. However, this choice is down to the company and whether they wish to invest in an officially signed certificate.

2.7.2 STRIDE Analysis

The STRIDE model is an acronym for Spoofing Identity, Tampering with Data, Repudiation, Information Disclosure, Denial of Service (DoS) and Elevation of Privilege. Howard et al [22] explain using this model to 'determine what tests to perform' following identification of security vulnerabilities. By applying this security testing strategy it will be possible for one to pinpoint areas of concern and resolve potential system compromise.

Such an approach should be taken into account not just during system testing, but also during the design and implementation stages of a project, as suggested by Efford [24]. Adhering to this would therefore lead a system possessing less security vulnerabilities. Howard et al [22] also mentions to test the expected successful result against the result the system outputs. This will be considered when defining the testing strategies for the system.

2.8 Usability Principles

Usability is defined by Dix et al [25] as how the user interacts with a product, as well as concerning issues on ease of usage and product acceptability. In addition, Preece et al [26] imply that a system that is not usable is not worth having. Jakob Nielsen [27] is a well known author in the usability field of computing. He has written a number of articles concerning the topic. Nielsen has defined a set of usability principles as guidelines for all new systems to follow.

With such information in mind, it will appear beneficial to incorporate usability into many aspects of the project. This can include the layout of a clear and consistent user interface, considering the users actions path within a system and modelling this into how the system navigates from one form to another. In addition some form of usability can occur during the testing and evaluating phases of the project, in order to check the system has met the users requirements and that the system works efficiently under the users control.

2.9 Project Amendments

Following the return of the Mid-Project Report during mid-January 2005, some significant revisions were made. The methodology was slightly readjusted in order to concentrate on reworking the implementation and testing phases of the project to aid rapid development. In addition, it was felt that the requirements already gathered were acceptable and reworking of all stages of the SDLC were not deemed necessary at the time.

The biggest amendments were changing the technology set used. Originally, PHP was chosen as the main scripting language. However, the requirements gathering had already taken place, and pointed towards a better tool and framework set to be implemented to cater for the proposed system. Not enough Security research was conducted in the mid-project report and considering the system is aimed to go live over the Internet, this research should have been done. The research performed is explained in Section 2.7.

3 Analysis

This chapter will gather the requirements for the proposed system to help the design stage incorporate these requirements successfully. Analysing sample documents helps understand how and what is recorded in the company, in order to model the system as close to the current business practices as possible.

Unified Modelling Language (UML) allows project designers to model requirements, workflow and stakeholders in the system. UML uses relatively simple graphics and symbols to explain situations and information flow. Johnson [28] explains using levels of abstraction to hide particular process and information can sometimes help define requirements. Another advantage of UML is it disregards features of different implementation platforms, allowing for effective concentration on the design requirements.

3.1 Current Systems Summary

Laurence Highman & Uniformity currently own two separate local area networks (LANs) within their London office. The first LAN is for an MS-DOS based accounts system, known as 'Cavalier', which is used to manage the company's account procedures. The 'Cavalier' system was a bespoke client-based package, written for the company over 15 years ago. It remains in use having, functioned flawlessly during this time period and is approved by their bank account provider. Enhancements to the system have been performed occasionally. Seven to eight client machines make up this peer-to-peer LAN, each workstation having its own peripherals, such as printers.

The second LAN runs an Administration server, which is used to manage their client database and marketing procedures. The server allows for the nine workstations to connect to it, currently all running

Microsoft Windows 2000 Operating System. Each workstation is relatively powerful, at today's comparisons, the lowest specification workstation being an Intel Pentium 4 with a 1.4 MHz processor. Aside from assigning access rights to each user, this server also hosts a Marketing Management System for the company, known as 'MMS'. This 'MMS' is a bespoke system to aid in the marketing and promotion of the company's products, in addition to aid in managing the present 5000 strong client base. The 'MMS' system inherits client-server architecture and has been in place for nearly eight years at present. Users also work with Microsoft Office 2000 Applications, in particular Word, Excel and Access. This second server also regulates internet access via a broadband connection. It is likely this connection will provide a means of access to the new web-based system.

3.2 Current Business Summary

Laurence Highman & Uniformity's business concerns the production and distribution of semi-structured career ware. They supply a variety of male and female clothing garments for industries from health and safety to civilian airlines. The company consists of two directors, who are also the joint business owners. The structure mostly resembles a hierarchical structure and four departments below the directors all sit on the same 'level'. Figure 3.2 shows an outline of the company structure and its departments.

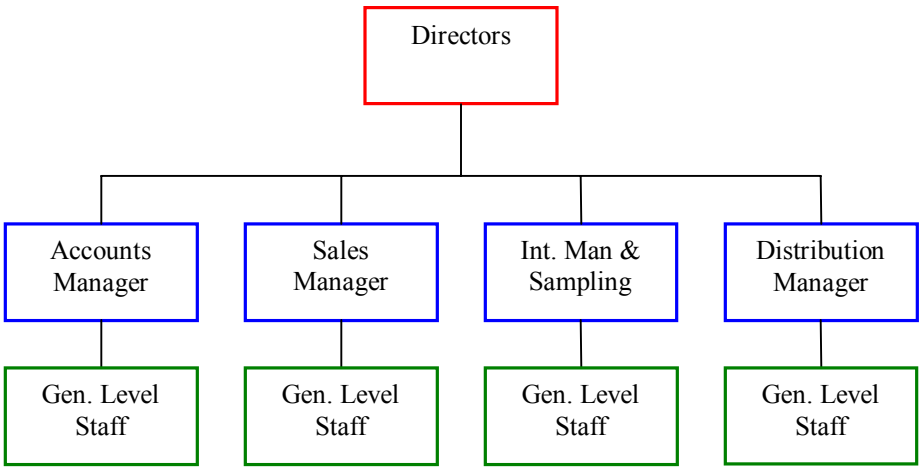


Figure3.2

The Accounts department is responsible for processing order payments from customers who have requested orders. This department is also responsible for providing payment to international suppliers of corporate garments, in particular uniforms. In addition staff are also responsible for chasing orders where payment has been delayed or not yet received. Suppliers will be contacted to refund payment for missing or under-quantified items if appropriate.

The Sales department is responsible for promoting clothing lines, mostly done in person. They are also obliged to gather order-lines for particular orders and in general, gather as many order-lines as possible for an order at any one time. This is due to some suppliers waiving, or reducing the total carriage cost for an order. However, this is entirely at the supplier's discretion and is generally dependant on what was ordered and the quantities ordered. Suppliers may also be contacted by Sales staff when orders contain missing or under-quantified order-lines.

The Internal Manufacture and Sampling Department is in charge of tailoring garments to the customer's requirements. For example, company logos may need embroidering on blazers and therefore, staff work machinery that performs the tailoring and alteration tasks. Occasionally, the company will also produce small bespoke samples for promotional purposes. The sampling manager must determine that the goods have met pre-defined company quality control levels. Orders are not business or time critical as order delivery dates are at the supplier's discretion, which Laurence Highman & Uniformity cannot control.

The Distribution department is responsible for gathering order-lines once they arrive at the London premises. They are also partially responsible for notifying the sales department of stock shortages for a particular item. Items need to be checked in on arrival and any order shortages or missing items reported to the Accounts and Sales Departments. This department prepare items for despatch to UK based clients. Appendix E illustrates a use case diagram, showing the tasks that the different departments perform. As supplier delivery dates are often uncertain, the number of orders arriving per day varies significantly.

3.3 Analysis of Current Business Processes

The previous Section defines the business and system structure. This Section aims to illustrate the detail of processes within the current system. Maciaszek [29] states that in order to understand requirements one must have a clear picture of how the processes are currently undertaken. Appendix F shows a UML activity diagram of the ordering process for the company. The diagram includes swim lanes to show the users involved at each stage.

Laurence Highman & Uniformity has developed a paper based system to cope with the compilation of orders. Each sales representative has their own purchase order book, where each page has four carbon copies (five copies in total). Orders may be spread over more than one order book, as many sales representatives can contribute to an order. In addition, when sales persons leave the premises to visit clients, their order book goes with them. Appendix D2 shows the current form used in the purchase order book.

The five copies are required to be distributed to the relevant departments. A white copy stays in the order book, a green copy goes to the Distribution department to check-in arrival of items, a red copy to the Accounts department, a blue copy to the archive and finally, a yellow copy to be faxed or posted to the supplier.

On creation of an order, the suppliers contact details, the order creation date and the date required by are recorded in each book. Appendix D1 shows each page has its own pre-printed order number and therefore, sales persons are adding order-lines to the same order number to for a particular order. Any specified delivery address other than the London offices would also be specified on this form. Sales directors add order-lines and record the quantity description, unit price and total for each order line. Sometimes sales persons add order-lines via phone/email, which proves time consuming. Once any order revisions have been completed, orders are ready to be processed. A Director examines each order book, checks the quantities and signs the order. The order is then faxed or posted to the suppler.

The supplier will send Laurence Highman & Uniformity a sales invoice (see Appendix D3) for the order-lines that were ordered and include carriage costs if appropriate to the order total. This is then processed by the Accounts department. Orders can arrive at any time and often depends on supplier punctuality. On arrival of goods the order-lines are ticked off the supplier's sales invoice and any missing or damaged goods are recorded here to be dealt with. Order-lines are then collated for each customer, and sent to the Internal Manufacture department who tailor the items to each customer's specifications.

3.3.1 Current System Shortfalls

From the above description, several shortfalls will need to be catered for in the new system. Firstly, order management becomes incredibly cluttered. Order-lines are added or removed and the scribbling out of such amendments is evident throughout all five copies of a purchase order and seems unprofessional when faxed to suppliers. Furthermore, as each sales person has their own order book and consequently duplicated order-lines can be ordered, which proves costly for Laurence Highman & Uniformity. Sales persons sometime manage orders in the nearest order book, if theirs is not available (if being signed by a Director). In addition when goods arrive it is difficult to see at a glance which order-lines are correct in their quantity. The final issue concerns security and fraud; orders can potentially be amended after being signed. This has occurred on one occasion where a previous employee had amended an order for their own personal gain. A computer-based system which prevents orders from being amended after being 'confirmed' would resolve this issue and hence, save the stakeholder unnecessary aggravation.

3.4 Feasibility Study

As discussed in Section 3.5, the client expressed the need for a solution that will improve performance in business process and processing orders in a more professional manner than the use of duplicate books. Hoffer et al [30] outlines a feasibility study to 'examine whether the development of a project is viable, particularly with relation to time and financial constraints'. Hoffer explains a number of categories to assess when performing feasibility studies. The categories that will be discussed in this Section are Technical, Economical, Legal, Operational, and Schedule Feasibility.

3.4.1 Technical Feasibility

According to Whitten and Bentley [31], this category concerns the practicality of the "proposed technical solution" in addition to accessibility of "technical skills, expertise and resources". With regards to resources, the majority of tools used for the solution development are open source. Apache Tomcat, Java 2 Software Development Kit (SDK), Java Server pages (JSP) usage and MySQL hold to the above statement by all being open sourced and readily available for anyone to use.

The client is prepared to invest in web hosting for the order processing system, as was mentioned in the Interview in November. The client also employs two staff members responsible for maintaining the two server systems currently in place. The in-house support staff would be prepared to support a system implemented using the tools mentioned above. The developer has also gained the skills at University to implement a system with the tools mentioned. Considering the knowledge of in-house support and experience gained by the developer at University, it has been concluded such a system would be technically feasible.

3.4.2 Economical Feasibility

Hoffer et al [30] makes emphasis on determining if a solution is economically feasible for the company. Furthermore, they recommend performing 'cost-benefit analysis' to achieve this. This is the process of gathering information to assess the benefits that the new system will offer against its overall cost. It is possible to define tangible and intangible costs or benefits. Tangible costs or benefits are far more evident as this can easily be measured in currency. Intangible costs however, cannot be measured easily as these may only be noticed over a long period of time.

Cost Analysis

As mentioned above the tools that run the application are cost free. In addition, the developer is implementing the system as part of a Final Year Undergraduate Project at University. The only cost of

development that were ensured by the client was time allocated to allow for interviews with the developer, allow the developer to gather sample documents and observe current business practices.

Some deployment costs were involved in the project. The solution would be hosted using an Internet Service Provider (ISP). The client decided to use Webconexion.net [32], as the client already had extremely good relations with this ISP for previously hosting a company promotional website. The client and developer discussed in the interview that this ISP may not necessarily provide the features required to host the system. However, after investigating prices with a number of other ISPs, the next cheapest found was Nameonthe.net [23] at £22.99/month, compared to £18.33/month for Webconexion.net [32]. Considering that Webconnextion.net was the cheapest available to offer the services required to power the system and the knowledge of good relations between the client and that company, the client decided to contact Webconnection.net to host the new system, once the system was ready for deployment.

In order for security to be achieved between communication from the hosting company and system users, an SSL certificate was created, as explained in Section 5.6. However, once the system will be deployed it would need a trusted third party to sign and verify the SSL certificate. After a telephone conversion with Webconexion.net on 22 February 2005, a sales representative informed me that an SSL connection could be purchased for £49.99/year. This would include the cost of a new certificate to be created and signed. This would allow for all connections between the users, whether in the London offices or mobile elsewhere to securely access the system over the World Wide Web.

Support costs need also be considered. Once the system goes live, Webconexion.net is responsible for maintaining the running of the MySQL database, Tomcat Web Server instance, and SSL connections. Any changes or support issues to the system that has been built by the developer would be undertaken by the existing in-house support staff. Mr. Rothschild was not keen on disclosing the cost of their support staff contract in their Interview. He did explain that as the support staff already takes care of two systems for the company, the extra cost of supporting another system would be marginal.

Benefits Analysis

While the costs have been clearly calculated, absolute measures of the new system's benefits are more difficult to quantify. Initially, one-time costs will have to be endured. This will include the gradual data migration from the current to the new system and user training / familiarity. However, as parts of the new system are to 'emulate the paper system' as discussed in the interview, the user's learning curve is likely to be less than expected, given most users already have internet usage experience. The system is also

likely to introduce other intangible benefits. For example, the system will deliver more timely information and therefore, the turnaround times to check-in items are expected to significantly decrease. However the main tangible benefits are likely to be cost reduction and overcharge avoidance, increased system flexibility and general human error reduction.

3.4.3 Legal Feasibility

Ayers [33] invites analysts to briefly consider the legal implications and feasibility of a new system. Laurence Highman & Uniformity will be expected to adhere to the Data Protection Act, 1998. Company Details will still be expected to be disclosed on request of that company and keep order and company records safe and secure. With the new system, records will be centralised and secure, rather than the use of several order duplicate books. In addition, validation implementation (see Section 5) is one method of preserving data integrity. It is also worth mentioning that Webconexion.net is also responsible for the safeguarding of company data on their servers. It is however anticipated from the telephone conversation with Webconexion.net and the good relations with the client, that Webconexion.net will adhere to the Data Protection Act and safeguard the information of the client. From this information, it has been concluded that the new order processing system for Laurence Highman & Uniformity will be legally feasible.

3.4.4 Operational Feasibility

Whitten and Bentley [31], defines operational feasibility as 'how the proposed system affects organisational structures, procedures and people'. The current system allows orders to be processed via telephone or email from external sales representatives and when directors work from home. However, with the new system a registered user may connect to the system and change the details of a particular order for example (providing that user has the access rights to do so). Therefore, this eliminates unnecessary telephone calls and email traffic, leaving the internal staff more time to perform the check-in of items when they arrive. Staff should also be able to allocate more time to generating new orders and therefore, increased revenue for the business.

The system's users will hopefully adjust to the new system quickly, as the reports generated will look similar to those already on the paper system. Familiar business terminology is used throughout the system, to minimise confusion and maximise user appeal. As mentioned above the users already have knowledge of using the internet and email. It is expected that with user manuals, online system help and general internet familiarity the users training will not be a learning curve that is too steep. It has therefore been concluded that the new system will be operationally feasible.

Other issues that could be considered under this category are cultural issues. Hollensen [34] explains that cultural issues of change may impact 'on the acceptability and adoption pattern of services'. While the cultural issues of change could be applied to an order processing system, it has been recognised to be beyond the scope of this report, and therefore only briefly mentioned.

3.4.5 Schedule Feasibility

Schedule Feasibility refers to how manageable the project is in terms of time available. A number of potential enhancements were suggested in the Mid Project Report dated 10 December 2004. Considering the time allocated to the project and these over-ambitious requirements, it was determined that the project plan seemed unrealistic (See Appendix B for Original Gantt Chart). To help reinforce a realistic schedule the developer needed to decide as whether to scale back the ambitious requirements to build an acceptable working system, or aim for a concept prototype, disregarding the system going live. The former was chosen, both to provide the client with a workable system and to give the developer rewarding real world experience.

3.5 Methods of Gathering Information

As assortment of information gathering methods are available to help define requirements. Interviews were found to be essential in understanding what the system needs to achieve, how to do so and shortfalls with current system practices and processes, according to Dimitrova [35]. Preece et al [26] implies benefits for imposing not just open ended and probing questions but a mixture of both in order to gain all possible views on the current problem situation. Some questions were pre-written but many arose during the interview as particular issues exposed their relevancy as suggested by Preece et al [26]. The interview took place with Mr. Michael Rothschild, one of the company directors. Furthermore, an Interview with the main director proved sensible due to the hierarchical business structure communication within the business may lack efficiency for common requirements tend to be shared by users. However, it was decided not to unnecessarily disrupt other staff, and observe them afterwards. It was therefore decided that other information gathering methods such as questionnaires would not have prove useful at this stage.

In addition to this interview, Mr. Rothschild agreed to the developer observing the users performing the current business practices. This ensured an improved understanding of the requirements and problems within the current system. Additionally, photocopied sample documents were kindly given to the developer during the observation process. Appendix D1 contains a sample Laurence Highman & Uniformity Purchase Order for 'Rael Brook Ltd', Appendix D2 shows an extract from a sales person's book and Appendix D3 shows the sales invoice received from 'Rael Brook Ltd' when the order arrives. Furthermore, appendices D4, D5 and D6 shows a Purchase Order, order book extract, and sales invoice

respectively, but for 'Regatta Ltd'. In addition, a Tagged Image File Format (TIFF) image file was emailed to the developer at a later date. This would be used when designing any electronic reports for the new system. These documents help reinforce the data required to be stored by the new system.

3.6 Requirements List

Requirements for the new system have been Sectioned into functional and non-functional requirements, in addition to technological or workflow constraints that have influenced these requirements. Johnson [36] contrasts between functional requirements as what the system needs to provide for its users and non-functional requirements, which concern functionality as opposed to specific system functions. The functional requirements have been split into features the system encompasses and obligatory constraints on system components. Johnson [36] identifies four groups which non-functional requirements can fall into, namely Usability, Security, Performance and Maintenance.

Data and Information Requirements
The system must be capable of storing a supplier's name, address, town, county, postcode, telephone and their fax number.
The system must store an order's id, code, delivery method, the date the order is required, the date the order was closed, any optional details, who confirmed the order and who the order is meant for.
The system must store an order line's id, item quantity, item code, size, colour, description, unit price, total, corresponding order id, which employee created the order line and who the order line is meant for.
The system must store a user's username, full name, company reference, telephone extension, access password and an access rights category assignment.
The system must store a number of actions that the system will allow and which type of user role can perform those actions. The system will therefore also have to store which users are assigned to a particular access role.
The system must store a list of order documents once an order has been confirmed. This will consist of the persistence of an id, the order number, the document and date of the created document.

Table 3.6.1

3.6.2 Functional requirements

System Features
Display an order summary page after a successful login. The system should be able to distinguish which orders are confirmed and unconfirmed, placing them in the correct view accordingly.
Provide a login screen, for which users will be validated entry before proceeding further into the system.
Users will be assigned to one of either 'Standard', 'Administrator' or 'Temporary User'. Users will be prompted with a message if their role prohibits a particular action from being performed.
To alert users of orders which are overdue, at a glance.
Warn before deleting company, user or order data, allowing users to cancel if desired.
To validate all fields when data is updated / created. This ensures data integrity by appropriately prohibiting null fields.
To provide efficient search facilities for locating suppliers, orders and customers.

Allowing one supplier to obtain many orders. However, one order line will be assigned to only one order.
No changes allowed to confirmed orders, conforming to the security issues mentioned by the Company.
Allowing order lines to be added and or deleted for a particular order. The order total should update automatically after order-line amendments.
Access rights should be able to be updated in real time; changes which should take effect without any reload or restart of the system.

Table 3.6.2

Constraints
Auto-generate and auto-increment the ID number for a particular record within the system, in particular for order and company tables, as these ID numbers are visible within the application.
Storage of address entries to a maximum of three lines, excluding the Town, County and Postcode fields.
The system should not allow an order to be confirmed where no order lines yet exist.
Enforce numeric integrity constraints on fields which must contain numeric data.
Enforce the entry of a correct date entry, in form of DD/MM/YYYY.
Storing a supplier, order or order line record if all the required fields are filled in correctly.

Table 3.6.3

3.6.3 Non-functional Requirements

Usability
Use of uniform colours and text layout throughout the system, maintaining consistency within the application.
Model reports closely to the appearance and layout of the current ordering system documents.
Use of contrasting colour and/or text where necessary, to stand out from the normal appearance of the working system, but without intrusive notifications.
Avoiding use of horizontal / vertical scroll bars, unless used within direct display of application on screen.
Positioning controls to prevent excessive time taken between mouse movements.
Show record updates to user in real time, by refreshing information with the updated information
Allow a user to cancel an action and ‘go back’ from wherever a user is within the application.

Table 3.6.4

Security
Prohibit users without a valid username and password combination from accessing any part of the system, barring a welcome page.
Restrict limits to the types of information particular parts of the system accept, in an attempt to strengthen hacker vulnerability.
Only allow administrators to view / change actions for particular role types that a user can be assigned to.
Preventing anyone other than administrators to remove or edit user’s credentials, including an update of an access role type.
Sending and receiving of all data via a secure channel, whether within the company LAN or over a Wide Area Network (WAN).

Table 3.6.5

Performance
The execution of user demands as rapidly and efficiently as possible.
The system should allow for concurrent execution of database queries, limiting the maximum number of concurrent transactions to 15 users.

Table 3.6.6

Maintenance
Allowing administrators to undertake simple routine housekeeping tasks, such as database backing-up.
Catering for other eventual developments in terms of functionality. The system should allow these further developments to be made with no disruption or effect to any already implemented portions of the system.

Table 3.6.7

4 Design

This chapter's purpose is to define the different components that make up the system and how they are connected together. The system uses the Model View Controller (MVC) pattern as defined by Geary [15]. This defines the architecture that the system adheres to. This chapter also defines and justifies the user interface and how the system would be deployed.

Data and requests are acted upon by applications. The pattern's purpose is to simplify the need for this to occur. Model represents data that the user expects to see. Normally, the model consists of Java Beans. The view renders the model. Normally, a view within a web-based application generates an HTML output which is interpreted at the client browser. Controller is responsible for processing and acting upon a user's requests, in order to manufacture an applicable model before passing the model to the view to be rendered.

Model View Controller Benefits

The MVC pattern aims to separate the presentation of content from its generation. The system will benefit from separating presentation and content as Poo and Kiong [37] explain. Therefore, it is possible to build the JSP pages and underlying Java code independently of each other. Any changes in one aspect of the MVC pattern do not affect other pattern aspects. This clear distinction of code separation allows for more seamless implementation and more straightforward debugging. For example, this allows for efficient testing of the Model layer.

4.1 Model

4.1.1 Java Database Connectivity

The model consists of a Java Database Connectivity (JDBC) Layer, which is used to execute SQL statements against the database. Reese [38], defines JDBC as an 'SQL Level API that allows you to embed SQL arguments to methods in JDBC interfaces'. Much like the MVC pattern itself, JDBC caters for a certain level of abstraction, as Reese [38] explains that one needs not know how JDBC routes SQL queries. JDBC offers benefits to the proposed solution as if the system should require a different database platform at a later date no changes need to be made to the database layer once implemented. Another benefit is that 'no configuration is required for network computers' as Sun Microsystems [39] states. This means software maintenance becomes centralised for the in-house support staff.

The system's JDBC framework is composed of four main classes. This framework wraps JDBC to make accessing the database even easier. It has been adapted from Java World [40].

SQLProcessor is used to update or query a table. The SQLProcessor executeUpdate() method executes a JDBC prepared statement, using a SQL string and a list of parameters. The SQLProcessor executeQuery() method works in a similar way, but makes use of the QueryProcessor interface to handle the query results.

The QueryProcessor interface defines one method called process. Classes which derive from QueryProcessor know how to construct application objects, based on specific JDBC result sets. Using polymorphism like this allows me to implement calls to the database with minimal amounts of code.

The PreparedStatementFactory class makes binding parameters to the JDBC prepared statement simple. Finally the PreparedStatementFactory class makes use of the NullSQLType class to allow null parameters to be bound to the prepared statement.

4.1.2 Prepared Statement Execution

As mentioned in Section 2.7.2, security should also be considered during the project's design phase. Efford [24], discusses methods on which an attacker can 'inject' SQL into systems to cause data damage. Reese suggests one method of overcoming this problem is usage of the JDBC 'PreparedStatement' interface. This binds parameters to set placeholders in a SQL statement. The placeholders are substituted with input parameters. The correct parameters must be passed into the statement in the correct order

otherwise execution fails. This is used as the system will be deployed over the internet to protect from potential hackers.

4.1.3 Database design

Appendix H models the requirements gathered from the Analysis phase of the project, in an Entity-Relationship diagram. This E-R diagram models the relationships between the entities to formulate a relational database schema prior to the process of normalisation occurring. In order to uniquely identify a record in an entity, each attribute within an entity is given a unique attribute, or unique attribute set where necessary. While entity-relationship diagrams will also depict entities and relationships between them, a class diagram is more appropriate due to the object oriented architecture of the system. The 'orderdocuments' table is deliberately not linked to any entity, as this plans to hold all orders on archive for life. Database operations on any other entity would therefore not affect the 'orderdocuments' entity.

In order to adhere to solid database design Elmasri and Navathe [5], state that database schemas must adhere to entity and referential integrity. These ensure that the primary key is never null and no unmatched foreign key values exist respectively. Entity Integrity is achieved by setting particular fields to 'NOT NULL' and Referential Integrity by recording the foreign key value for the primary key in the referring table. Using low level detail from the Analysis project phase, the schemas can be designed now to include, entities, attribute details and relationships between them. The schema must not only model the current system but any other requested enhancements. Details of the entire schema designed for the solution has been tabulated in Appendix K.

Table Normalisation

According to Date et al [41], the benefit of Normalisation is to ensure that no data is duplicated within a schema. Adhering to a particular 'normal form' ensures a level of consistency and data integrity will be achieved. Four different normal forms are commonly used to normalise system schemas, with Boyce Codd Normal Form (BCNF) adhering to the strongest constraints, Date et al [41]. It can be concluded that all tables in the solution schema adhere to third normal form. This states that any attributes dependant on each other must be separated, to be dependant upon the 'super key' for that class.

4.2 View

4.2.1 Java Server Pages Design

The purpose of the Java Server Pages (JSP) is to render the object contents it receives. Geary [15] recommends instructing the view to retrieve the most current data when required. This would achieve system consistency, by updating views immediately after records change. A validation JSP page will also

be included. This page contains detailed error messages and only appears in the main form if any validations are thrown by the application layer. Finally, 'hidden' html fields will be used on some pages to store a record ID but not display it. Geary [15] suggests to use this feature to pass the ID back to the application layer to perform some operation. Using this advice, the ID can be used to delete or retrieve records by its unique field when passed to the application layer. This also ensures referential integrity is withheld and the relevant process occurs on the correct record.

4.2.2 Java Server Pages Design

iText is an Open Source Java Library and therefore freely available by Lowagie [42]. The library contains numerous classes and methods to paint Portable Document Format (PDF) files. The library does not just persist PDF files to hard disk store but allows creation of 'PDF documents on the fly', Lowagie [42]. This means that the 'PDFWriter' can call methods within the system's architecture to populate the page with the relevant order-lines for example. This system will be used to generate the order reports for the company. Its methods will allow a PDF document to be 'painted' to the layout of the existing paper system and therefore, confirming to the client's requirements.

Furthermore, IText's capabilities extend to be able to write file streams for a number of file formats to include, Microsoft Word, HTML and Rich Text Format (RTF) files, Lowagie [42]. This would mean that the company would be able to expand the system in the future if desired, possibly to include customised Microsoft Word letters to automatically mail suppliers over order-line quantity disputes.

4.3 Controller

The framework, acting as the Controller for the application is made up of four main files. An 'action' servlet will handle URL requests ending in '.do' forwarding requests to 'action' java beans. Although not intended , the '.do' extension Geary [15] suggests, will help mask the identity of scripting language that the system operates with. Actions defined within the system will implement the 'Action' interface, an 'ActionFactory' creates action instances, the 'ActionServlet' maps action requests and the 'ActionRouter' directs requests to the appropriate JSP page. A sequence diagram for the ActionServlet is shown in figure 4.3. The framework wraps an 'actions.properties' file which is used to store a mapping of requests to actions. The framework and sequence diagram 4.3 are both taken from Geary [15], but have been adapted to suit the requirements of the system.

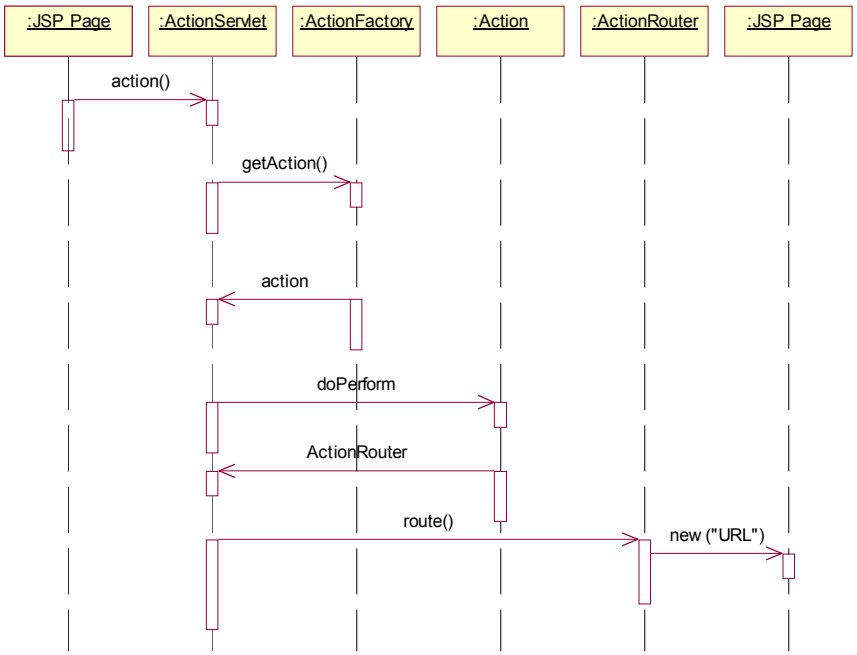


Figure 4.3

In the system the controller handles process request and responses, before populating the JSP page (the view aspect). MVC promotes code-reuse as once the entire pattern has been implemented for the first object in the system, the addition of other objects follow the same style of implementation. The system will therefore benefit from code consistency and providing the initial development pattern is error free, it will never have to be revisited. Appendix I shows the framework for the system.

4.4 User Interface

For some form of consistency to remain present through out the application, a template was devised to define the layout of the interface. Figure 4.4.1 shows the template created. The browser caption located near label 1 will change according to which part of the system the user has navigated to. When a user has logged into the system the header near label 2 will display a user's login credentials. When a user is not logged in the company name will only be displayed. Nielsen [27] suggests that the user should not be able to access parts of the system that are not appropriate to them. Therefore, the sidebar near label 4 will customise the number of links available to a user dependant on their access profile. Each page body will be shown in the area near label 5. Help will also be available on each screen with a help icon near label 7. This design makes use of JSP includes.

4.4.1 Usability Design

Using Nielsen's usability principles [43] also states consistency in system colours to represent the different features available. Such consideration enables users to gain system familiarity. Table headings

and the sidebar will be sky blue to match the company logo colour. Furthermore, Nielsen [43] implies the use of contrasting colours where necessary. All errors are red to stand out against a white background. Using colour contrast in this method allows for users to quickly read data off screen. Create Retrieve Update Delete (CRUD) actions are medium orange, help and information displays are green and finally, links within the sidebar are white to indicate separate parts of the system, to allow rapid selection of system features. Cascading Style Sheets (CCS) support will be incorporated into every system page. Specifying format attributes in one CSS file makes all changes global (to all pages importing the style sheet) as suggested by Castro [44]. This also allows for interface consistency throughout the system and easy modification if required.

All buttons within all forms will be of the same size, and all forms containing text boxes will enforce tab indexation. Required fields are also marked with a '*' to minimise users receiving validation error messages. The only part of the display which changes is the main area near 5. All other parts of the screen remain static. The sidebar links have been deliberately itemised with the most common system features in the middle. The login/logout link always remains at the top and less common features (such as Access Rights) are at the bottom of the list of links.

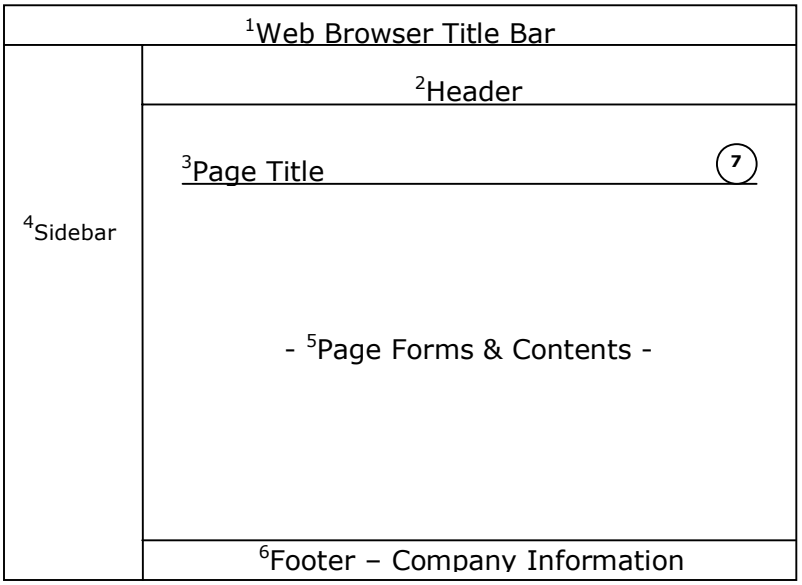


Figure 4.4.1

4.4.2 System Navigation

Appendix J shows a navigation map that users can follow to perform various system functions. The names of all sidebar links, buttons and record actions have used company terminology, wherever possible. For example 'finish order' refers to an order that has been delivered and checked-in. This will aid rapid user familiarity. The navigation path a user follows anywhere in the system aims to mirror the sequence

of processing orders outlined in Section 3. Users start with a welcome screen displaying a logo (not shown in Appendix J) before proceeding to the login screen. From here users are routed to an order summary page (shown in a green line in Appendix J). Items shown in grey indicate system functions available to administrator users only.

4.4.3 JSTL Design

Bergsten [20] defines the JSP Standard Tag Library (JSTL) as an extension to the JSP language by enabling development of custom actions. In addition, Bergsten [20], explains the relevance of JSTL usage within the MVC architecture by allowing views to change regardless of the other layers. This effectively means that the system can easily undergo future enhancements to the view without affecting any other part of the pattern. The company could request an extension for each user to be able to customise layout and content presentation of pages. Such an implementation could therefore be easily achieved. JSTL was chosen over scriptlets as shown by Sun [45], as both Scriptlets and JSTL allow usage of custom actions. However, JSTL's syntax is faster to apply and easier to read, which will speed up project development.

4.5 Component and Deployment Design

Figure 4.5 shows a component and deployment diagram showing the physical deployment of the implementation components. Bennett et al [46], implies such diagrams benefit to plan a system's architecture as well as paths of communication between the physical hardware nodes. All users within the office already have internet access via their existing Windows network. However, as the system will be deployed on the Internet, all users whether in-house or not access the system via an SSL/TLS encrypted channel over a WAN (the Internet). Figure 4.5 illustrates that the web hosting server will support JSP, JSTL, MySQL Database and Apache Tomcat. The client machine only requires the Operating System (OS) and Web Browser as the JSTL runs within the JSP pages themselves.

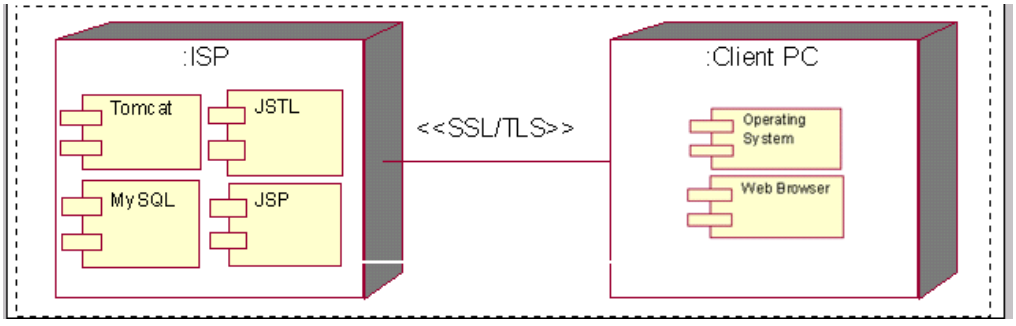


Figure 4.5

5 Implementation

The purpose of this chapter is to document the tasks performed in order to implement the design as described in Chapter 3. This chapter documents the sequences of operations performed in order to implement a system that would meet the requirements. The chapter will discuss database, server, application and security implementation that occurred during this phase of the project.

5.1 Database Implementation

The primary phase of implementation was creating the database and its contained tables. As discussed in Section 2, MySQL was the chosen database technology. This tool allowed for the straightforward creation of schema and all tables, constructed via an inbuilt shell window.

5.2.1 Tables

In total ten tables were created that relate to the system's functionality. When creating tables in MySQL, each table is assigned a name and the table fields are specified. Each field specified contains its own name, a primary key if appropriate, data type and field length.

Primary Key Applications

Most tables were assigned with a primary key that was set to `auto_increment`. The primary key would ensure record integrity by uniquely identifying each record entry within the system. The 'auto_increment' flag assigns integers in sequence to new records, aiding integrity maintenance as suggested by Atkinson [47]. The system user need not worry about maintaining primary keys for the orders, orderlines, company and user tables. The role and access tables do not use `auto_increment`, as these tables contain hard-coded data on **all** the possible access and user-type combinations allowed by the system.

The `role_access` table contains a 'composite key' relating to the primary keys in role and access tables respectively. In this way, all the possible actions allowed by each of the three user types as defined by the requirements, would be persisted to this table. The composite key is two or more columns designated together as the primary key. Similarly, the `ordercheckin` table contains a composite key relating to the primary keys in order and orderline tables. In the `orderline` table one order can have many order-lines. Therefore the `orders_id` and `orderline_id` fields were combined to form a composite key, to store data concerning item whereabouts on arrival to the premises.

Data Types

The most common data type used within the system was the VARCHAR, which permits persistence of alphanumeric data items. These were used for telephone fields as well to allow numbers to be entered

with space characters between them for ease of reading. For all data requiring date/time to be persisted, the DATE data type was used to maintain integrity. The DATETIME data type was used where a time needed to be stored, such as archiving orders in the `ordercheckin` table. Price information used the DOUBLE data type, with a format of (7,2). This imposes all prices to have a maximum of two numbers after the decimal place. All primary and composite keys were unsigned, preventing a key from ever being assigned mathematical operators (-1 for example). The Binary Large Object (BLOB) data stores the data stream to output archived orders as a PDF file. A standard MEDIUMBLOB was used allowing 16 Megabytes as a maximum length, sufficient enough for the text and images used by this system.

Other Imposed Constraints

An entry within the code field of the orders table will not always be of the same length. The order code consists of the customer and order number in the form '1/1'. This field was therefore left restricted to ten characters. This code format eliminates the need to generate codes manually. Many fields in the system should not be left empty in order to maintain data integrity. Such fields were flagged NOT NULL, to ensure that the database required data to be present for those particular fields. Furthermore, some fields were initialised with a default flag, being set if no data was persisted for that field. For example, when an order is confirmed, corresponding entries for the appropriate order-lines are made in the `ordercheckin` table, with an initial comment field for that order set to a 'null string'.

Referential integrity constraints between tables as defined in Section 4, is upheld by referencing primary and foreign keys within each relevant table in the form **`FOREIGN KEY (`***`field_name`***`) REFERENCES `*****`parent_table`*****`(`***`field_name`***`)`**. If a particular action violates this constraint type, such as deleting a record in the parent table, MySQL will generate an error. Such relationships may either me one-to-one or one-to-many. Logical Deleting has been used on the user and company tables. If for example, an employee is dismissed from Laurence Highman & Uniformity, but has contributed to an unconfirmed order, the system performs a logical delete. This marks the deleted field for that user, but does not physically delete the record. The system determines at a later date when the user can be deleted. Some MySQL syntax demonstrating constraints and some data types discussed is shown below:

```
CREATE TABLE orderdocument (id int(10) unsigned NOT NULL auto_increment,
order_no varchar(45) NOT NULL, doc blob NOT NULL,
date datetime NOT NULL, PRIMARY KEY (id))
```

5.2 Server Implementation

As mentioned in Section 2 the chosen web server was Apache Tomcat. The Tomcat administration manager was used to specify the port number and hostname that the entire application will run on. These credentials were matched with those already supplied to the database properties.

Enhancements to the Tomcat Configuration

A small number of additions were made to the `web.xml` file in the Tomcat directory, which is read upon a server start-up. As discussed in Section 4, the MVC framework does not support the ability for an action to revert back to another action before proceeding to the JSP page. Therefore a servlet-mapping was introduced which would declares an extension for an action to route to another, if required. All actions in the client's application end with the extension `.do`.

```
<servlet-mapping>
   <servlet-name>action</servlet-name>
   <url-pattern>*.do</url-pattern>
</servlet-mapping>
```

The servlet used in the system servlet handles all requests to a browser request(s). The system specifically handles requests in servlet mapping, ending with .do In order that all actions would actually route to another action, the web server requires the Java package to be specified, containing the class for servlet routing. This class tests whether the action ends with the extension `.do` and routes a request to another action or JSP page as necessary.

```
<servlet>
<servlet-name>action</servlet-name>
<servlet-class>com.adamraymond.framework.ActionServlet</servlet-class>
      <init-param>
            <param-name>action-mappings</param-name>
            <param-value>actions</param-value>
      </init-param>
</servlet>
```

The session timer was implemented by adding the following line in the same file, to make a users session timeout after 20 minutes of inactivity.

```
<session-config><session-timeout>20</session-timeout></session-config>
```

5.3 Client Application Implementation

Section 4 mentions the framework used for the client's application (MVC). The next stage was to enhance the existing framework and to build functionality into the application. This Section focuses on the implementation of business processes and functional requirements as defined in Section 3. Non-functional requirements such as usability are discussed further on in this chapter.

5.3.1 Laurence Highman & Uniformity Framework

Categorising Action Types

Actions have been divided into two clear functional groups. Display Actions which retrieves data from the database to populate the JSP page as the user requested. In addition process actions which process modifications to the database that the user requested. In order to utilise the ability to route from one action to another, the application defines actions within two Java packages. The process package contains actions that must perform operations or process data from a request. The display package contains actions that gather the required data in order to correctly populate the JSP page. By implementing this, actions can be routed from one type to another or even the same action type, prior to being routed to a JSP page. For example, updating a company record requires the record's data to be displayed by a `DisplayAction`, the updated fields to be retrieved and validated by a `ProcessAction`. Finally the updated company record would be displayed via another `DisplayAction`, before proceeding to the JSP page.

The system uses four main files on which the application is built upon as mentioned in Section 4. One small modification made was testing if a user was still logged into the application in the `Action.java` class. If the user is not logged in they are routed to the login page. Each action extends this class, and tests for user login presence. This syntax utilises session-timeout functionality as described above.

```
if( loginRequired && !isUserLoggedOn( req ) ){
      return new ActionRouter( Routers.LOGIN_PAGE ); }
```

Other Validation Types

Each 'action' class calls its `getRequestParams` method, which obtains the user's input at the JSP page, if any. The action's `doPerform` method may be called to perform any operations required, for example saving user's input into an object to persist to the database. However if the user's input returns unexpected parameters the framework will throw a Java exception. Custom exception-handlers were written which could be thrown application-wide, returning sensible error messages back to the JSP page (and user). The syntax below calls methods in `Validator.java` class in the `framework` package. Here, a type-check prohibits text being entered into a field requiring numeric data.

```
if( ! Validator.isLong( quantity ) ) {
      validateErrors.add ( new String
      ( "Please enter a valid entry for quantity." ) ); }
else  { orderLine.setQuantity( new Long( quantity ) ); }
```

Aside from the type-check like that expressed above, a vast number of validation checks were implemented system wide the syntax above builds a list of error messages, which the user gets to see all in one go if more than one error is thrown. This is displayed in way presentable to the user.

Range-checks were also used when comparing dates entered. For example, a user cannot search for an archived order beyond the current date

Constants Implementation

The application implements a set of interfaces defined within the `constants` package. Interfaces allow unrelated classes bearing similarities to interact, without imposing class relationships. Four interfaces are defined for the application. `ActionCodes` contains a list of all possible actions within the application, used to verify access codes against the database. The Routers interface defines action names and corresponding action URLs. The `Parameters` interface defines names of attributes for objects entering and leaving requests from JSP pages. An `Object` interface allows parameters to be assigned to a particular object type, particularly useful when more than one parameter type populates a JSP page. For example, the order summary screen is populated from company, order and user objects, all within the same screen.

5.3.2 Java Database Connectivity

As mentioned in Section 4 Java Database Connectivity (JDBC) is used to connect the Java application to the MySQL database. The JDBC layer is coded with the database address in order to successfully execute queries and table updates. The `jdbc` package contains classes with methods to execute queries, execute updates and add parameters to a SQL statement. The `PreparedStatementFactory` class allows SQL strings to be written, specifying `?` wherever parameters are inserted. Parameters are passed in to replace question marks, build SQL statements and execute them before returning results collections. An example of Prepared Statement syntax is shown below to update a user.

```
private static String UPDATE = "update user set username = ?, fullname = ?,
reference = ?, extension = ?, password = ?, accessrights = ? where id = ?";
```

QueryProcessor's are classes which convert known results set into a collection of data transfer objects. For every query made in system a QueryProcessor derived class converts the results set returned from Database into the DTO. For example, querying the company table a result set it built. The syntax below indicates this particular example.

```
public Collection process( ResultSet rs ) throws SQLException
 {
     Collection queryResults = new ArrayList();
     while( rs.next() )
     {
         CustomerDTO customer = new CustomerDTO();
         customer.setId( new Long( rs.getLong( "id" ) ) );
         ----------------------------------------
         customer.setDeleted( new Boolean(rs.getBoolean("deleted")));

         queryResults.add( customer );
     }
```

```
        return queryResults;
    }
```

5.3.3 Report Generation

As mentioned earlier in this chapter the `orderdocument` table uses a BLOB object to store confirmed orders as a byte stream. Once orders are confirmed they may no longer be amended. The iText library was used to create an order template that all confirmed orders would be constructed from, in the form of PDF documents. When orders are confirmed by an administrator, this class gathers required information for an order (order-lines, order and company credentials), populating the pre-defined layout, to create the byte stream. This byte stream is inserted into the `orderdocument` table, along with the order code. The syntax below is a small portion of how the PDF document is created. The report writer in writes it in way that defines layout of all text , positioning entries on report font size etc. orderReportPDF takes database structures to populate the PDF view in order to supply the dynamic data.

```
// Supplier Address Line 1
Phrase phraseAdd1 = new Phrase( "", times10 );
Cell cellAdd1 = new Cell( phraseAdd1 );
cellAdd1.setHorizontalAlignment( Element.ALIGN_RIGHT );
cellAdd1.setBorder( Table.BOTTOM );
tab.addCell( cellAdd1 );
```

When users view an order, the appropriate byte stream is read and a PDF file is created in the 'temp' directory of the application. An Adobe viewer window is then launched to view the relevant order. It is important to note that when the next order is viewed, the same PDF file is updated in the 'temp' directory (called 'TempOrder_<session_no>.pdf') gets session number from user session, to cater for concurrency and therefore the order viewing window. While many entries may exist in the `orderdocument` table, only one PDF file ever exists on the server.

5.3.4 Item Check-In Monitoring

As mentioned in Section 3, some orders do not arrive complete. Order-lines may arrive with quantities above or below the requested amount, or not at all. When an order is confirmed, an entry in the `ordercheckin` table is made for all the order-lines for a particular order. Each entry contains the `order_id`, `orderline_id`, `quantity` and `comment`. The quantity field is set to zero and the comment field is null for each entry at this stage. When a confirmed order is checked-in the order's order-lines are displayed. The user can change the quantity received field for each individual order-line (from its original value of zero) and add an optional comment. When such information is updated, the application must determine whether the quantity, comment, or both fields were updated for all the order-lines in the view. The `UpdateCheckInProcessAction` class handles this by determining the name

of the returned parameters and applying a `substring` at particular string indexes (see partial syntax below). For example, "comment_2_9" indicates to update the comment field where `order_id` = 2 and `orderline_id` = 9.

```
String commentOrReceived = value.substring ( 0, value.indexOf("_") );
String orderID = value.substring( commentOrReceived.length() + 1, value.lastIndexOf("_") );
String orderLineID = value.substring( value.lastIndexOf("_") + 1, value.length() );
Iterator it = tempCheckIns.iterator();
```

5.3.5 Report Search Facilities

Once orders are finished, only their entry in the orderdocument table remains. The search facilities within the application allow users to find archived orders by using one of three selection criterion in a particular time interval. A search by 'Sales Person' returns all orders that a particular sales person has contributed to. A search by 'Customer' returns any orders that a customer had between a given time period. Similarly a search by 'Supplier' returns all orders that a supplier contributed to in a given time period. The system returns the order code, and confirmed date for each successful hit, along with the opportunity to view archived orders as a PDF file.

5.3.6 Persisting Updated Information

The application uses Data Transfer Objects (DTOs) to hold the parameters returned from the JSP page in one Java Bean. For example, if a user's credentials are being updated (assuming the system user does not press 'Cancel') the `UpdateUserProcessAction` calls `getRequestParams`, setting the user's object with the relevant parameter values.

Data Access Objects (DAOs) are used to perform an operation with the contents of the DTO bean. Using the above example of updating user information, the `updateUser` method is called on the `UserDAO` object, taking the DTO bean as a parameter. This method then updates the appropriate record in the database via the `UserProcessor` (query processor). The `UserProcessor` knows which fields to update, as its parameter names are the same as those parameter names passed from the DTO. This syntax shows the call to the DAO method when the 'OK' button is pressed on the page.

```
if( !isCancelled() ) {
    UserDAO.updateUser( conn, user); }
router = new ActionRouter( Routers.VIEW_USERS_DISPLAY_ACTION );
```

5.3.7 Retrieving Data for Display

The application's Display Actions call DAO methods to query data from the database. Once the application determines the user has access to that part of the system, the DAO populates the DTO bean

with all the attributes that are returned by the retrieve method. Following this the DTO is used to set the parameters into the request that a JSP page will use to display the query results. The action then routes to the JSP page that will display the data concerned. This syntax is part of the `ViewUserDisplayAction` to display a user's details.

```
determineAccess (conn, req, AccessCodes.USER_RETRIEVE);
UserDTO user= UserDAO.retrieveUserById( conn, id );
req.setAttribute( Objects.USER, user );
router = new ActionRouter( Routers.VIEW_USER_PAGE );
```

5.4 User Interface Implementation

The user interface has been built using JavaScript, Java Server Pages, Cascading Style Sheets (CSS), and HTML. Together, these techniques have built functionality, validations and usability into the application. This Section discusses these technologies in relation to their implementation.

HTML Layout

The application view is created HTML from JSP code execution. In effect, each JSP page consists of HTML code with the JSP syntax inserted at specific points. This allowed for pages to follow a set layout and general appearance while displaying the correct information from the database. For example when viewing a user's credentials, the database output of the username was shown by:

```
<td class = text>${user.username}</td>
```

Where a URL needs parameters passed into it, the JSP code is placed within the HTML anchor. For example, in order to update a user's details the URL would be of the following format.

```
<a class = actions href ="update-user-display action.do?id=${user.id}">Update</a>
```

JSP Implementation

Each JSP page contains an 'action' tag, specifying a Process Action. This Process Action handles the parameters the JSP page passes to it. Some JSP pages also encompass a hidden variable normally used to store a record ID, but is never displayed on screen. This variable is sometimes passed to the back-end of the application where an ID is needed to retrieve a record, for example. As the value for the ID is unique in the context of the table required, the application ensures that the correct record will be retrieved.

```
<form action='<%= response.encodeURL("update-checkin-process-action.do") %>' method="post" >
```

Validation operations are also executed within JSP pages. For example, `updateCheckIn.jsp` tests the quantity of each order-line against the quantity received. When a different quantity received for an order-line is entered, the JSP page changes a helper image on that order-line reflecting the new quantity. An example of this is shown when more items than required are sent for an order-line.

```
<c:if test="${CheckInFullDTO.orderCheckIn.qtyReceived >
CheckInFullDTO.orderLineFull.orderLine.quantity}">
      <td align='center'>${CheckInFullDTO.orderLineFull.orderLine.quantity}
      <img src="images/arrow-up.gif" width="15" height="15"  >
</c:if>
```

JavaScript Implementation

JavaScript excerpts were implemented throughout the application to launch popup windows for the help system. On each page there exists an icon of a 'green man' with the phrase 'Help' to launch the appropriate HTML help file. JavaScript was also used to launch the popup window when viewing an order in the form of a PDF file.

```
function viewHelp() {
      url = "html/help/viewOrdersSummaryHelp.html";
      window.open (url, "newWind", "status,resizable=1"); }
```

CSS Implementation

As mentioned in design, CSS was used to maintain consistency of the system's appearance. The best example of CSS in the system is row_counter usage. This sets each entry in a JSP table with a number. Then code was written to make odd and even rows different colours.

```
<%
if( java.lang.Math.IEEEremainder( row_counter, 2 ) == 0 )
 { out.println("<tr class=colourtablerow>"); }
 else { out.println("<tr class=nocolourtablerow>"); }
 row_counter += 1;
%>
```

5.6 Security Implementation

Security has been applied on all possible part of the system. Text boxes have been limited to take set lengths of input by defining Maxlength = n on each text box where 'n' matches the field length in the database structure. The 'post' tag on all actions hide the attributes returned to the view that can display in the address bar. The system also ensures that even if users have been logically deleted, they cannot login to the system.

Based on framework in design (Section 4) it was required to generate all specific action classes. On top of the general framework involving action classes was the implementation of the access control system, It is based on interrogating user currently logged into session. The method interrogates request to check if the user is logged in.

```
determineAccess (conn, req, AccessCodes.ORDERLINE_CREATE);
```

The SSL certificate was created by creating a new keystore from scratch in Java's standard key store format, and allows one to specify where the key will be stored locally.

```
keytool -genkey -alias tomcat -keyalg RSA
-keystore \path\to\my\keystore
```

6 System Testing

The aims of the testing procedures deployed in this project are to ensure that the system functions as intended by the developer. It is important to note that system testing contrasts from a system evaluation. The former refers to whether or not the system has achieved its requirements set out. The latter refers to how well its requirements were achieved. Conduction of the testing phase partly occurred during the system's implementation. By doing this, individual system components were assessed as well as overall application functionality. This chapter discusses the use of realistic sample data, testing methods undertaken and their results.

6.1 Sample Data Deployment

The sample data was gathered from data stored the current manual system. In total six users, ten companies, three confirmed orders, three unconfirmed orders, and pre-set access rights data were used to populate the system's database. In addition, each order had at least three order-lines associated to it, as well as special order instructions in some cases. This allowed for the system's functionality to be examined and help simulate a real world environment, particularly if the system is to work with the current data as an eventual data migration plan.

6.2 Unit Testing

The purpose of Unit Testing is to ensure that each unit in the system functions properly. This testing strategy was applied to each object in the system. A Java test class containing methods to assess objects, calling each method with an assortment of parameters to test whether the values returned were appropriate. These test classes ensured that valid input data was accepted and invalid data was rejected. Each unit within the application was tested to verify that all links and buttons navigated as expected. Usability testing has also been incorporated to examine whether error messages are clear and understood. The database schema was also monitored to ensure insertions, updates and deletes were occurring and with the expected changes. The results for Unit Testing can be found in Appendix M. All the tests have proved successful. For those tests which failed initially, the causing factor was identified and remedial action was taken in order to pass the test concerned. The detail of remedial action taken has been documented in Appendix M. Security testing has been incorporated to verify that the expected result is in face, the actual result.

6.3 Black Box Testing

The purpose of Black Box or functional testing is to assess the system's internal workings. However the overall aim of this test strategy is to examine results without knowing how the system arrived at that result. In effect, a tester only requires knowledge of the system specification rather than underlying architecture. This caters for such testing to be performed from an end user's perspective rather than a designer's perspective. Furthermore, any ambiguities that may exist between the Black Box test results and the original specification are easily detectable, as an unexpected output would occur. The results of Black Box testing can be found in Appendix N. The tests performed here are based around a user's input and actions. The expected and actual system output for each test case has been documented.

6.4 White Box Testing

The purpose of White Box testing is to certify that the underlying system architecture functions correctly. This contrasts to Black Box testing which examines system output from knowledge about the system's use of syntax. White Box testing has been undertaken for this system, in order to examine the changes of state for each of the database tables within the system. White Box testing has two immediate benefits. Firstly, by examining the table states during update, create and delete processes for example, the developer is able to verify that the right tables are being queried or affected. Furthermore, the developer can query the record contents, to ensure the correct and relevant fields in one or more tables have been correctly queried or affected, as appropriate. The second benefit is more aimed at the client; as such testing can simulate system stress that it may endure once deployed to see how it reacts. The results of the White Box testing can be found in Appendix O.

6.5 Integration Testing

The purpose of performing Integration Testing is to ensure that the different portions that make up the system function correctly, when combined to form a single working application. Appendix I shows the implementation phases, the 6 phases of implementation were defined. Integration Testing was performed at the end of each phase to prove that the system still maintained functionality. The benefits become more apparent as the system increases in size and functionality. For example, at the completion of phase two, Companies and Users were two separate system entities (at that point in development). However, the addition of Access Controls in phase three relies on successful integration with the Users entity, for login and customisable access rights to be successful in their operations. For such reasons, testing occurred at each phase's completion to verify that new additions have not affected existing functionality. Navigation from one part of the system to another was also examined, by testing the links within the system's sidebar menu. As mentioned in Section 5, the implementation of CSS helps to uphold a consistent layout format

for each page in the system. This was also examined. The outcome of Integration Testing for each phase of the system is shown in table 6.5. This phase was executed until no international errors were found.

Phase	Test Case	Success?
1	No other phases to test any impact for.	N/A
	No other phases to test any consistency against.	N/A
	Sidebar links to phase one entity works correctly.	Test Successful
2	Addition of Phase 2 has no impact on existing Phase 1	Test Successful
	Addition of Phase 2 does not cause any inconsistencies. All Phases are functioning appropriately.	Test Successful
	Sidebar links to phase 1 and 2 entities work correctly.	Test Successful
3	Addition of Phase 3 has no impact on existing Phases 1 and 2.	Test Successful
	Addition of Phase 3 does not cause any inconsistencies. All Phases are functioning appropriately.	Test Successful
	Sidebar links to phases 1, 2, and 3 entities work correctly.	Test Successful
4	Addition of Phase 4 has no impact on existing Phases 1 - 3.	Test Successful
	Addition of Phase 4 does not cause any inconsistencies. All Phases are functioning appropriately.	Test Successful
	Sidebar links to phases 1, 2, 3 and 4 entities work correctly.	Test Successful
5	Addition of Phase 5 has no impact on existing Phases 1 – 4.	Test Successful
	Addition of Phase 5 does not cause any inconsistencies. All Phases are functioning appropriately.	Test Successful
	Sidebar links to phase 1, 2, 3, 4 and 5 entities work correctly.	Test Successful
6	Addition of Phase 6 has no impact on existing Phases 1 – 5.	Test Successful
	Addition of Phase 6 does not cause any inconsistencies. All Phases are functioning appropriately.	Test Successful
	Sidebar links to phase 1, 2, 3, 4, 5 and 6 entities work correctly.	Test Successful

Table 6.5

6.6 User Acceptance Testing

In order to determine that the system had met the requirements defined in Section 3, the user and developer had arranged to meet at the end of March 2005. This involved the developer demonstrating the software and the directors using it briefly. User acceptance testing was done at the end of the implementation The developer provided a form which stated the original requirements. The results of this testing strategy can be found in Appendix Q. As this does not provide a robust conclusion to whether the software was still potentially usable, Section 9 discusses User evaluations that were also performed.

7 Evaluation

This chapter revisits the minimum requirements and assesses how these were achieved. In addition, methodology and user evaluations are given in addition to self-criticism on the system produced.

7.1 Meeting the Minimum Requirements

Summarise the current corporate clothing systems and business.

Using a semi-structured interview in November allowed the developer to gather information regarding the current business structure, how the business has evolved and therefore, the recognised need for a better working system. In addition sample documents and user observations allowed the developer to understand the current systems specifications and ordering process.

Perform a feasibility study of the proposed solution.

The feasibility study in Section 3 discusses a variety of categories concerning the new system's feasibility. Volumetric information was discussed in the interview concerning the number of users in the system and the number of orders arriving daily. In addition, criticality of orders was discussed in order to build a picture of how feasible the new system would be.

Produce a database solution incorporating web-based access.

The Design Implementation chapters of this report document how the web-based order processing was designed and implemented. In addition, Appendix L documents a number of system screen shots. Once the framework for the initial application was in place, implementing the enhancements to the system was manageable. Exploiting class inheritance and code-reuse that Java offers, allowed for rapid development and consistency of the system. Needless to say, the extra functionality became more complex as implementation progressed.

Compose a User Manual to compliment the solution.

Following the interview in November and after performing the requirements analysis for the new system, it had become evident that certain parts of the system would not be used by anyone other than an Administrator. Efford [24] recommends that users should not need to see parts of the system not relevant to them. With this in mind, two separate operating manuals were composed, one for all users and the second for Administrators. These were defines as deliverables in Section 1 of this report.

Perform a basic evaluation of the new system.

As each new phase was implemented, it was reworked to iron out any bugs or errors. The new system has met the minimum requirements and other enhancements. The old system relied on every sales person having their own order book and therefore composing one order from a number of sales person's was awkward and complicated. The new system was welcomed as it centralises the view of all orders and reduces errors by enforcing field validation. Mobile costs have been reduced significantly.

Mr. M. Rothschild (the main director) implied that in-house costs could potentially be reduced with the new system. Laurence Highman & Uniformity sometimes manages up to 110 orders/month. Fax costs to suppliers could be sliced from an average of £127 on fax communications to £45 per month. The implementation of a Digital Signature would allow this to occur, as orders could be emailed securely to other suppliers.

7.2 Exceeding the Minimum Requirements

Table 7.2 shows the list of enhancements that were implemented

No.	Functionality
1	The user login, logout and online help systems.
2	The ability to create a facility for archiving previous orders, for future retrieval.
3	Secure Sockets Layer (SSL) encryption and security certificate.
4	Customisable access rights, assigning users to role types.
5	The management of partially (and complete) despatched orders.
6	The monitoring of missing and delayed items from suppliers.
7	Calculating order and order-line totals in real time, immediately updating system view.
8	Substitution capabilities; missing stock is replaced for similar in an order.
9	Standard User Manual, designed to explain everyday features.
10	Administrator User Manual, for advanced system features.

Table 7.2

7.3 Potential System Enhancements

7.3.1 Determining Carriage Costs

Some supplier's state a minimum amount the order must reach for carriage to be free. It would be useful to record this amount with company details. The system could then alert users when the order becomes reaches or exceeds this amount, and print something to the effect of 'Carriage Free' on the PDF order form faxed to the supplier.

7.3.2 Improved Administrator Features

Instead of forcing directors to use a tool called MySQL Administrator to perform a backup or restore of the database's contents, a more user friendly interface could be built in to the system. A useful feature here would be allowing Administrators to perform a 'One Click Backup' to a specified file location.

7.3.3 Generating Order Quotations

A useful feature for a sales person would be to quickly generate an order quotation. This would be under a separate part of the system which would be similar to creating an order, but would automatically generate a quotation reference number valid for a pre-defined time period. A planned order can then be 'activated' when the time becomes necessary.

7.3.4 Customisable Interface and Page Layout

At present data displayed in the system adheres to a pre-defined sorting type. It would be beneficial to users to be able to sort results by any column heading by clicking on that column. Furthermore, allowing users to customise page layout and colour schemes improves overall system usability for each user.

7.4 Evaluation of Chosen Technologies

From a consistency perspective, CSS implementation allowed all pages to adhere to the same textual and layout format. Tweaking the properties in a single CSS file meant changes were global to the entire system and therefore timesaving. On reflection, the JSP page implementation was also favoured by the developer. The use of JSTL rather than scriptlets (as explained in Section 4) allowed for neater display of data. JSP tags were considered to be far 'easier on the eyes' during the implementation phase and the developer felt JSP tags were a superior choice, particularly when debugging the system.

The Java Platform catered for relatively seamless connectivity between the MySQL database and JSP views. Parts of the system exploited code re-use and method inheritance from the framework, explained in Section 4. In addition where the framework did not cater sufficiently for a specific problem, methods were overloaded in the child classes via the exploitation of polymorphic syntax.

MySQL database allowed for all the requirements to be fulfilled. However, MySQL does not currently support views. Therefore this downfall required the explicit creation of displaying particular fields each with a specified 'alias' and resulted in rather complex queries. In terms of future expansion, MySQL supports transaction and row locking to be easily added to the system. Transactions ensure that SQL commands are executed with atomicity (execute all or nothing). If errors occur the transaction 'rolls

back'. Additionally, the usage of composite keys does slow execution down slightly with MySQL; Atkinson [47].

7.5 Evaluation of Chosen Methodology

As mentioned in Section 2, the chosen project methodology was a variation of the SDLC. This catered for the implementation and testing phases to be reworked using an iterative approach. This approach proved useful in these phases, as debugging the system while implementation occurred was probably less costly than finding bugs at the end of development. However, the analysis and design phases were dwelled on to ensure that all the requirements were being catered for. It may have been better to also rework these phases to ensure that requirements were met while possibly saving more time for development. Reworking the analysis and design phases would therefore have been slightly more geared to the projects requirements.

The original and revised Gantt charts are illustrated in Appendices B and C respectively. Originally, the design phase would not start until end of January. However after further research by the end of December 2004, it was realised that a more suitable but similar design framework was more appropriate, and implementation would need more time than originally planned. Implementation began mid-January immediately after examinations.

7.6 Evaluation of Implementation

The initial implementation of each of the 3 architectural layers (Presentation, Application and Data) was the most meticulous task. This required successful operation between, the database, Java implemented framework and JSP pages. Once this had been achieved for one object in the system, implementation of the remaining objects involved similar coding practices, which by this time the developer was more familiar with.

One part of system implementation that was slower than anticipated was database table revisions. As more objects in the system were added some of the table structures changed. An example of this was the addition of a 'deleted' column for `user` and `companies` tables to allow logical deleting to overcome foreign key constraint. This simply set a record's deleted field to '1' and therefore the JSP page test not to display records which are marked deleted. On the next application load the system will delete the record from the database permanently, if the user is not assigned to any non-finished orders.

Overall the implementation was a smooth phase of the system's development. The implementation was broken down into phases (as shown in Section I), which helped the developer concentrate on one phase at

a time. This ensured that the developer can test the system so far at each stage before developing the next phase of the system, as well as ensuring that all requirements were met. However, it is worth noting that White Box testing is somewhat determined by the accuracy of the tester themselves.

7.7 User Evaluation

Appendix Q shows the results of the user evaluations. In total five users were asked to evaluate the system towards the end of April 2005. Each user had the same set of tasks to perform under the new and old system. The tables in Appendix Q show the results of each user performing the tasks set. The results have shown that overall the tasks performed under the new system is mostly faster than that compared to the current system. The tasks undertaken by the five users were creating a new order, updating an existing company, creating an order-line and searching for a previous order.

Creating an order produced slightly varying results compared with the current system whereas updating an existing company showed faster results compared to the current system. Creating an order-line was marginally slower than the original system but searching for orders was significantly faster in the new system. It is anticipated that more use of the software will increase operational familiarity, as far as users are concerned. Therefore the time taken to create order-lines should decrease with more system usage.

Users provided verbal feedback of the system after the test procedures. All users were generally pleased with the system's look and feel and that controls were well laid out. One distribution manager particularly favoured the use of icons to determine which order-lines were above, below, or matching their required quantity, mentioning that at a glance it is possible to see an order-line which could pose a potential problem.

The directors were generally pleased with the outcome of the system, in particular with the production of PDF documents when an order is ready for processing. They felt that as other companies also send them electronic documents as sales invoices, they felt confident is responding which documents that looked neat and professional.

The sales managers liked the idea of being able to view and manage details away from the premises when required. Although when the system was demonstrated via a URL that accepts an SSL connection, the system was slightly slower, taking nearly 2.5 seconds extra to respond to some actions. Sales managers appreciated the ability for the system, to automatically generate order codes, which is something that they

are required to enter manually with the current system. In addition, the automatic calculation of order totals would mean no more untidy amendments on each order book.

Sampling workers also provided feedback on the system. They commented that the system does not appear to benefit their everyday practices. However, this was expected to be the case as they were not intended to be users of the system. A sampling manager however commented that he might benefit from a way of recording which garments have been altered or finished tailoring. However this has been considered beyond the system's scope, as the system concerns processing orders and not goods outgoing to customers.

Overall the client was impressed with the new system's features. All sales persons wholeheartedly agreed that centralising the order management means no more time wasted conferring order details from each sales person's order book. Mr. M. Rothschild expressed that the company would not be prepared to migrate their data to the new system as such a move would be a big gamble. He implies in a letter that the company would consider the deployment of the system within the next 6-9 months (see Appendix S).

7.8 System Criticism

7.8.1 Improved Concurrency Controls

In its current state the system does not support full concurrency management. Section 5 explains how currency was implemented when many users view records. However to allow all users to work with the system without causing data integrity problems, full concurrency controls would be implemented given more time. This could be achieved by adding a 'version' field to each table in the system. Say when two users view records the version number is '1'. If one user updates a record the version number becomes '2' for that record. When another user attempts to edit the same record, the version number had changed between the time viewed and an update requested. This error can be caught and displayed to users.

7.8.2 Logging and Auditing Enhancements

Garfinkel and Spafford [48], suggest two internet security enhancements. Logging concerns logging all errors occurring in a system to a log file when exceptions are caught. The file can then be manually interrogated**.** Auditing allows administrators to see which parts of the system users have been into at a very high level. The sequence of actions a user does forms the order trail. The results could be filtered by action or user.

7.8.3 JavaScript Enhancements

While testing it was noticed that on the systems access page, if administrators repeatedly press the 'OK' button impatiently, to save changes the page can cause an error, showing no entries to be persisted. While this is somewhat benign and a very rare occurrence, it could be solved with some JavaScript to disable the button after a single press.

Bibliography

[1]. Avison, D.F., G., ***Information Systems Developments: Methodologies, Techniques & Tools, 3rd Ed.*** 2003: Mc Graw Hill. pp. 39-576.

[2]. Avgerou, C.C., T., ***Developing Information Systems: Concepts, Issues and Practice, 2nd edition***. 1998: Macmillan. pp.58-71.

[3]. Hughes, B.C.M., ***Software Project Management, 3rd ed.*** 2002: Mc Graw Hill. pp. 5-79.

[4]. Measureit.com. ***Measureit.com***. [cited 01 February 2005]; Available from: http://www.measureit.com/images/SDLC.gif.

[5]. Elmasri, R.N., B., ***Fundamentals of Database Systems, 3rd Edition***. 2000: Addison Wesley Longman. pp. 11-18.

[6]. igrep. ***Upgrading Access to a multi-user environment***. 2005 [cited; Available from: http://www.aspfree.com/c/a/Microsoft%20Access/Upgrading-your-Access-Application-for-a-Multi-user-Environment.

[7]. db-review.com. ***Listing Supports: Script Editor***. 2004 [cited 01 December 2005].

[8]. Wiley. ***Why MySQL?*** 2004 [cited 30 November 2004]; Available from: PostgreSQL.

[9]. Harrington, J., ***SQL Clearly Explained***: Ap Professional. pp.43-48.

[10]. Microsoft Corporation. ***SQL Server: How To Buy***. 2004 [cited 01 December 2004]; Available from: URL: http://www.microsoft.com/sql/howtobuy/default.asp.

[11]. Sklar, D.T., A. (2003), ***PHP Cookbook***: O'Reilly. pp. 16-54.

[12]. Tanenbaum, ***Computer Networks 4th Ed.*** 2003: p. pp. 618-623.

[13]. Corporation, M. ***Internet Information Services***. 2004 [cited 01 December 2004]; Available from: URL: http://www.microsoft.com/WindowsServer2003/iis/default.mspx.

[14]. Project, J. ***SSL: How to***. [cited 17 December 2004]; Available from: http://jakarta.apache.org/tomcat/tomcat-4.0-doc/ssl-howto.html.

[15]. Geary, D., ***Advanced Java Server Pages and Servlets***. 2001: Prentice Hall. pp. 72-184.

[16]. Quigley, E., ***Perl By Example***. 1998: Prentice Hall. pp. 26-39.

[17]. Gray, N., ***Web Server Programming***. 2003: p. pp.168-197.

[18]. Gutmans, A., ***PHP 5 Power Programming***. 2004: p. pp.39-58.

[19]. Greenspan, J.B., B., ***MySQL/PHP Database Applications***. 2004.

[20]. Bergsten, H., *Java Server Pages*. 2003: O' Reilly. pp. 4-78, 525-621.

[21]. Whyte, W., S., ***Enabling E-Business***. 2001: Wiley. pp.99-103.

[22]. Howard, M.L., D., ***Writing Secure Code, 2nd Edition***. 2003: Microsoft Press. pp. 572-577.

[23]. ***Nameonthe.net*** *(21 February 2005) URL: http://www.nameonthe.net/hosting.jsp*.

[24]. Efford, N., D., ***SY32: Secure Computing***. 2005: University of Leeds.

[25]. Dix, A.F., J. & Beale, R., ***Human - Computer Interaction***: Pearson. pp. 5-7.

[26]. Preece, J.R., Y. Sharp, H., ***Human-Computer Interaction***. 1994.

[27]. Nielsen, J., ***Usability Engineering***. 1992: Academic Press. pp. 238-245.

[28]. Johnson, O., ***IS21: OO Design and Analysis***. 2003.

[29]. Maciaszek, L., ***Requirements Anaysis and System Design***. 2001: Addison Wesley. pp. 17-53.

[30]. Hoffer, J.G., J. and Valacich, J., ***Modern Systems Analysis and Design***. 2001: Pearson. pp. 72-100.

[31]. Whitten, J.L., and Bentley, L. D., ***Systems Anaysis and Design Methods 4th ed.*** 1998: Irwin McGraw Hill. pp. 720-727.

[32]. ***Webconnexion.net Web Hosting Packages*** *(20 February 2005) URL: http://www.webconexion.net/services/java_web_hosting.php*.

[33]. Ayers, R., ***The Essence of Professional Issues in Computing***. 1999: Prentice Hall.

[34]. Hollensen, S., ***Global Marketing: a decision oriented approach***. 2004: Pearson. pp. 73-74.

[35]. Dimitrova, V., ***Questionnaires and Interviews for the FYP***. (2004), University of Leeds.

[36]. Johnson, O., ***IS23: E-Commerce Information Systems***. 2003.

[37]. Poo, D.K., D., ***Object Oriented Programming In Java***. 1998: Springer. 112-134.

[38]. Reese, G., ***Database Programming with JDBC and Java***. 1997: O' Reilly. pp. 6-62.

[39]. Sun. ***JDBC Overview***. 2004 [cited 25 December 2004].

[40]. JavaWorld. ***Eliminating JDBC Overhead***. 2005 [cited 03 January 2005]; Available from: http://www.javaworld.com/javaworld/jw-05-2002/jw-0524-sql.html.

[41]. Date, C.D., H. & Lorentzos, N., ***Temporal Data and the Relational Model***. 2001: Elsevier. pp. 38-41.

[42]. ***iTEXT***. [cited 01 March 2005]; Available from: http://www.lowagie.com/iText/.

[43]. URL, N.J. ***Jakob Neilsen's Ten usability heuristics***. 2002 [cited 29 November 2004]; Available from: http://www.webreview.com/1997/10_10/strategists/10_10_97_2.shtml.

[44]. Castro, E., ***HTML With XHTML and CSS.*** 2003: Peachpit Press. pp. 131-136.

[45]. Sun. ***Scriptlets***. 2005 [cited 10 February 2005]; Available from: http://java.sun.com/products/jsp/tags/11/syntaxref11.fm5.html.

[46]. Bennett, S.S., J. & Lunn, K., ***Schaum's Outlines of UML***. 2001: Mc Graw Hill. pp. 270-276.

[47]. Atkinson, L., ***Core MySQL: The Serious Developer's Guide***. 2002: Prentice Hall.

[48]. Garfinkel, S.S., G., ***Practical Unix & Internet Security***. 1996: O' Reilly. pp. 161-166.

Appendix A – Project Reflection

When starting the project, I was definitely over ambitious as to what requirements could be implemented. I had to choose between building a software prototype and scaling down the requirements to something more realistic given the limited time available. I chose the latter to build something that would actually be used. This allowed me to apply knowledge and concepts learnt at University, gaining real world experience in solving a problem. My advice to future students is not to be over-ambitious regarding requirements and beware project development may take longer than originally anticipated.

In my opinion I have developed my skills in both technical and non-technical areas. From a non-technical perspective I would say that 'the questions asked in interviews are almost as important as the solution designed'. The Professional Development modules (PD15 and PD31), provided guidance on how to present myself and communication skills as a Computing professional; to dress appropriately and deal with clients in a friendly but professional manner. My advice to students conducting interviews is to have some clear questions in mind regarding business structure, processes and practices prior to the interview. The client appreciated I had a clear idea what information I aimed to gather. Asking more probing questions to the client from their responses as the interview progresses then became easier.

From a technical perspective the software development of the project drew on expertise of technical issues learnt throughout my University study. Having developed software resembling 3-tier Client architecture applying these skills to a real world problem has helped strengthen my software development skills in a number of areas. However, having now completed the project I now understand how important a methodology is to a project schedule. If undertaking such a project again I would be stricter on myself concerning project management, as I somewhat dwelled on analysis and design phases. My advice is read background information and read it well. I had to change my design framework in late December after realising a different technology set was more relevant to the problem.

Overall, I have thoroughly enjoyed the project from the outset. My final words of advice: 'Enjoying is half the battle. Do not choose areas of the course for your project that are either irrelevant or unappealing to you'. While it has been challenging it has also been equally rewarding. I aim to implement the remaining further enhancements to the client's software and now understand and appreciate the importance of good requirements gathering and project management, which partly dictates project success.

Appendix B- Project Schedule (Original Gantt chart)

ID	Task Name	Start	Finish	Duration	Nov-04 1/11/04	8/11/04	15/11/04	22/11/04	29/11/04	Dec-04 6/12/04	13/12/04	20/12/04	27/12/04
1	Define Problem, Aims & Requirements	01/01/2004	14/11/2004	2 wks									
2	Regular meetings in London	01/01/2004	21/11/2004	3 wks									
3	Interviews with Sales Representatives	08/11/2004	14/11/2004	1 wk									
4	Interviews with Managing Director (MD)	08/11/2004	14/11/2004	1 wk									
5	Research into Methodologies & Technnologies	15/11/2004	28/11/2004	2 wks									
6	Determine Methology & technologies	22/11/2004	05/12/2004	2 wks									
7	Mid-Project Report	22/11/2004	12/12/2004	3 wks									
8	Interviews - MD / Sales Rep's	13/12/2004	09/01/2004	4 wks									
9	Learning Selected Technologies	20/12/2004	06/02/2004	7 wks (int.)									
10	Feasibility Study	24/01/2005	30/01/2005	1 wk									
11	System Investigation & Analysis	24/01/2005	06/02/2004	2 wks									
12	System Design	31/01/2005	06/02/2005	2 wks									
13	Implementation: Internal System	07/02/2005	06/03/2005	4 wks									
14	Implementation: External System	07/02/2005	06/03/2005	4 wks									
15	System Testing	07/03/2005	20/03/2005	2 wks									
16	Produce User Manuals	21/03/2005	04/04/2005	2 wks									
17	Deploy System	28/03/2005	03/04/2005	1 wk									
18	Evaluation of System	04/04/2005	17/04/2005	2 wks									
19	Final Report Write-Up	21/03/2005	25/04/2005	7 wks									

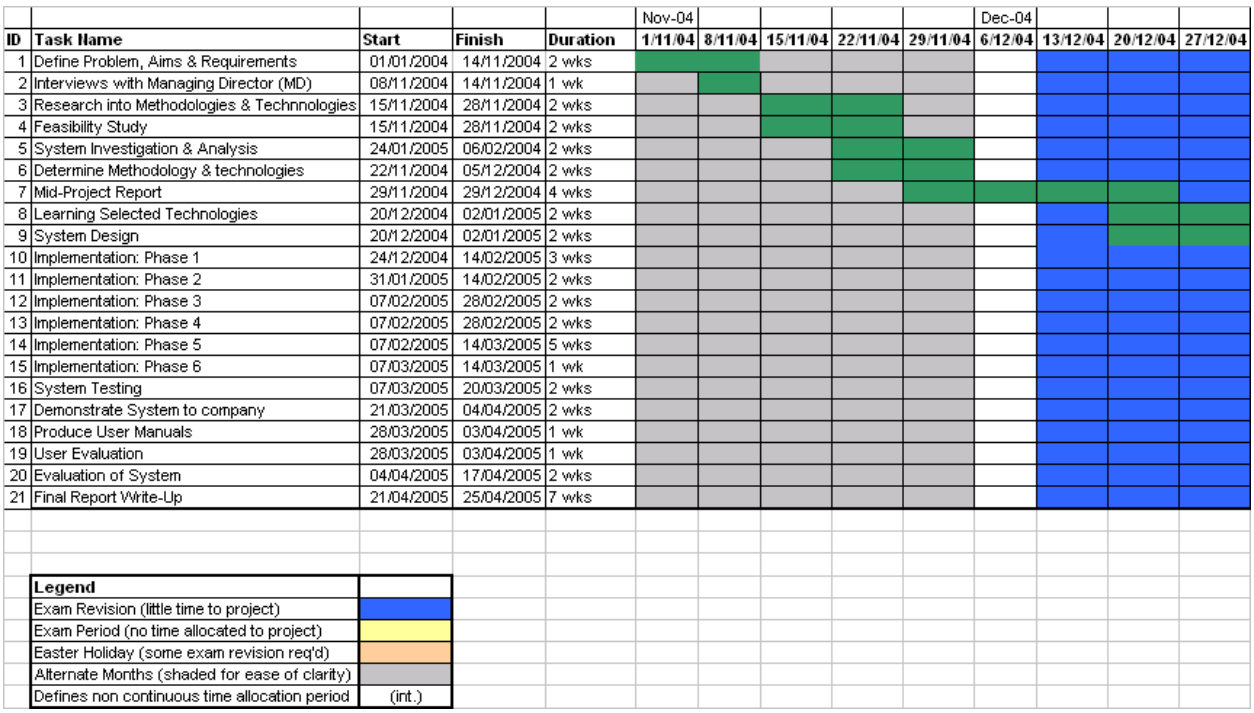
Legend	
Exam Revision (little time to project)	
Exam Period (no time allocated to project)	
Easter Holiday (some exam revision req'd)	
Alternate Months (shaded for ease of clarity)	
Defines non continuous time allocation period	(int.)

Continued

Jan-05 3/1/05	10/1/05	17/1/05	24/1/05	31/1/05	Feb-05 7/2/05	14/2/05	21/2/05	28/2/05	Mar-05 7/3/05	14/3/05	21/3/05	28/3/05	Apr-05 4/4/05	11/4/05	18/4/05	25/4/05

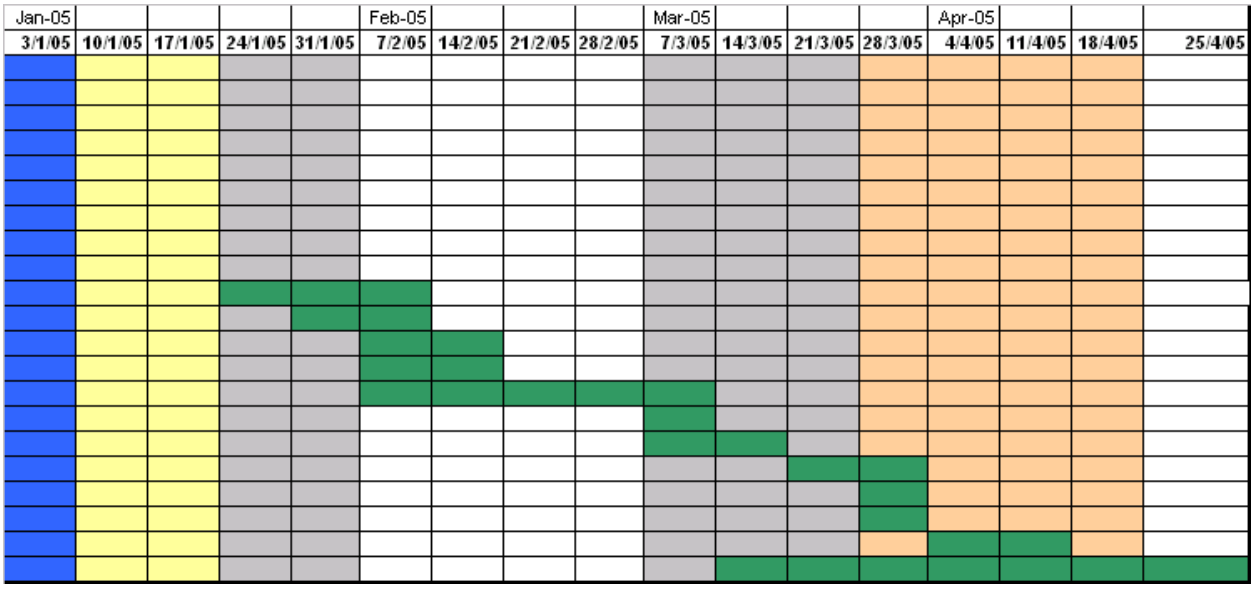
Appendix C - Project Schedule (Revised Gantt chart)

ID	Task Name	Start	Finish	Duration
1	Define Problem, Aims & Requirements	01/01/2004	14/11/2004	2 wks
2	Interviews with Managing Director (MD)	08/11/2004	14/11/2004	1 wk
3	Research into Methodologies & Technnologies	15/11/2004	28/11/2004	2 wks
4	Feasibility Study	15/11/2004	28/11/2004	2 wks
5	System Investigation & Analysis	24/01/2005	06/02/2004	2 wks
6	Determine Methodology & technologies	22/11/2004	05/12/2004	2 wks
7	Mid-Project Report	29/11/2004	29/12/2004	4 wks
8	Learning Selected Technologies	20/12/2004	02/01/2005	2 wks
9	System Design	20/12/2004	02/01/2005	2 wks
10	Implementation: Phase 1	24/12/2004	14/02/2005	3 wks
11	Implementation: Phase 2	31/01/2005	14/02/2005	2 wks
12	Implementation: Phase 3	07/02/2005	28/02/2005	2 wks
13	Implementation: Phase 4	07/02/2005	28/02/2005	2 wks
14	Implementation: Phase 5	07/02/2005	14/03/2005	5 wks
15	Implementation: Phase 6	07/03/2005	14/03/2005	1 wk
16	System Testing	07/03/2005	20/03/2005	2 wks
17	Demonstrate System to company	21/03/2005	04/04/2005	2 wks
18	Produce User Manuals	28/03/2005	03/04/2005	1 wk
19	User Evaluation	28/03/2005	03/04/2005	1 wk
20	Evaluation of System	04/04/2005	17/04/2005	2 wks
21	Final Report Write-Up	21/04/2005	25/04/2005	7 wks



Legend	
Exam Revision (little time to project)	
Exam Period (no time allocated to project)	
Easter Holiday (some exam revision req'd)	
Alternate Months (shaded for ease of clarity)	
Defines non continuous time allocation period	(int.)

Continued



Appendix D – Current Documentation

Appendix D1 – Sample Purchase order for 'Rael Brook'

Lawrence Highman & UNIFORMITY

CORPORATE WEAR, DESIGN & MANUFACTURE

4-10 ST. ALPHAGE HIGH WALK
LONDON EC2Y 5EL

Telephone: 020 7628 0165
Facsimile: 020 7628 2945

Deliver to the above address unless otherwise indicated here

RAEL BROOK

Purchase order

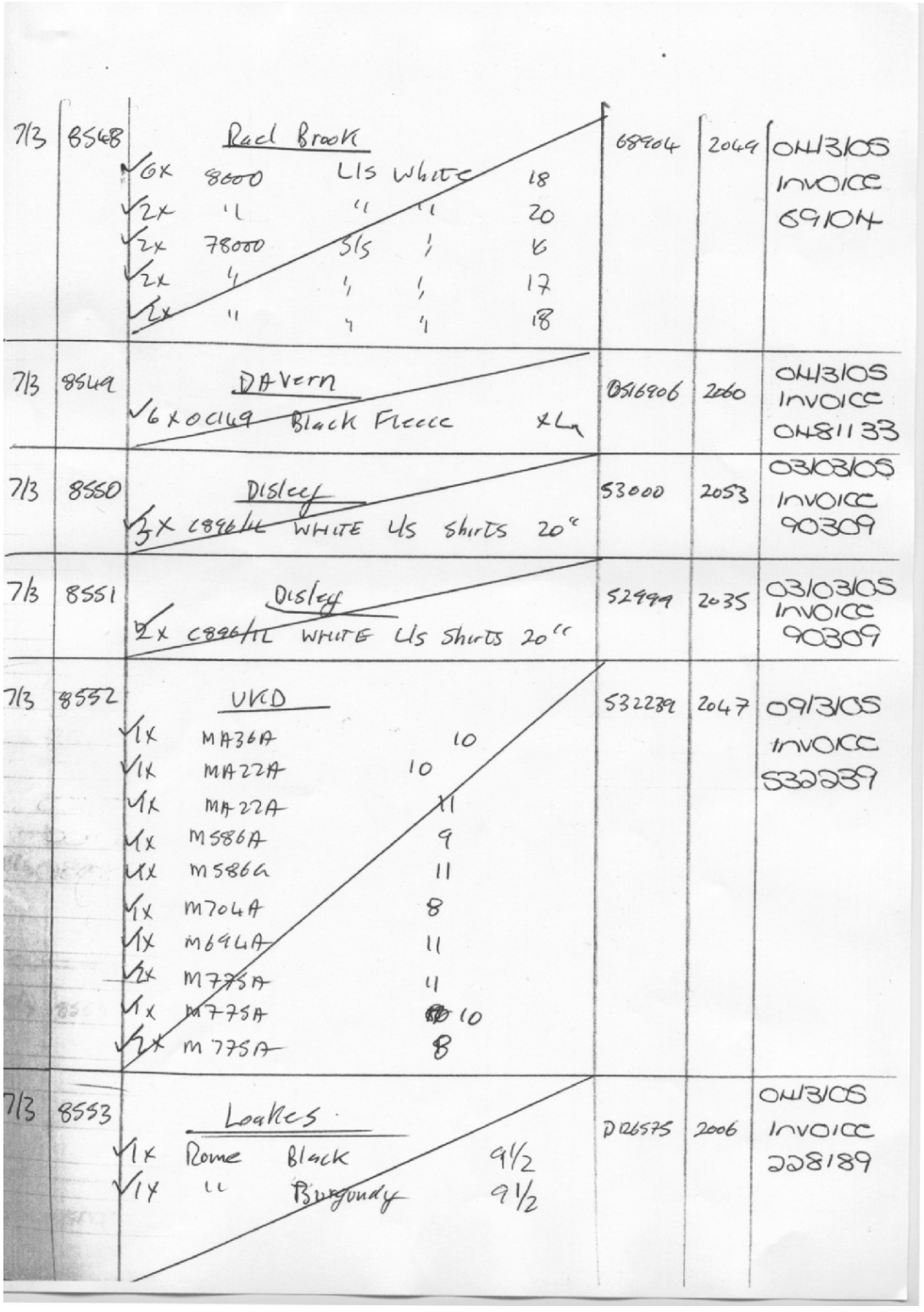
Goods in Number			
Goods to be sent by	Date required: ASAP	Date of order: 03.03.05	Order number: VAR /2049/

Quantity	Please supply the following	Unit price	Amount
8 ✓	8000 s/s white shirts 6 - 18" 2 - 20"	5.00	✓
6 ✓	78000 s/s white shirts 2 - 16" 2 - 17" 2 - 18"	4.50	✓
	14		

Our order number must appear on all invoices, delivery notes etc.

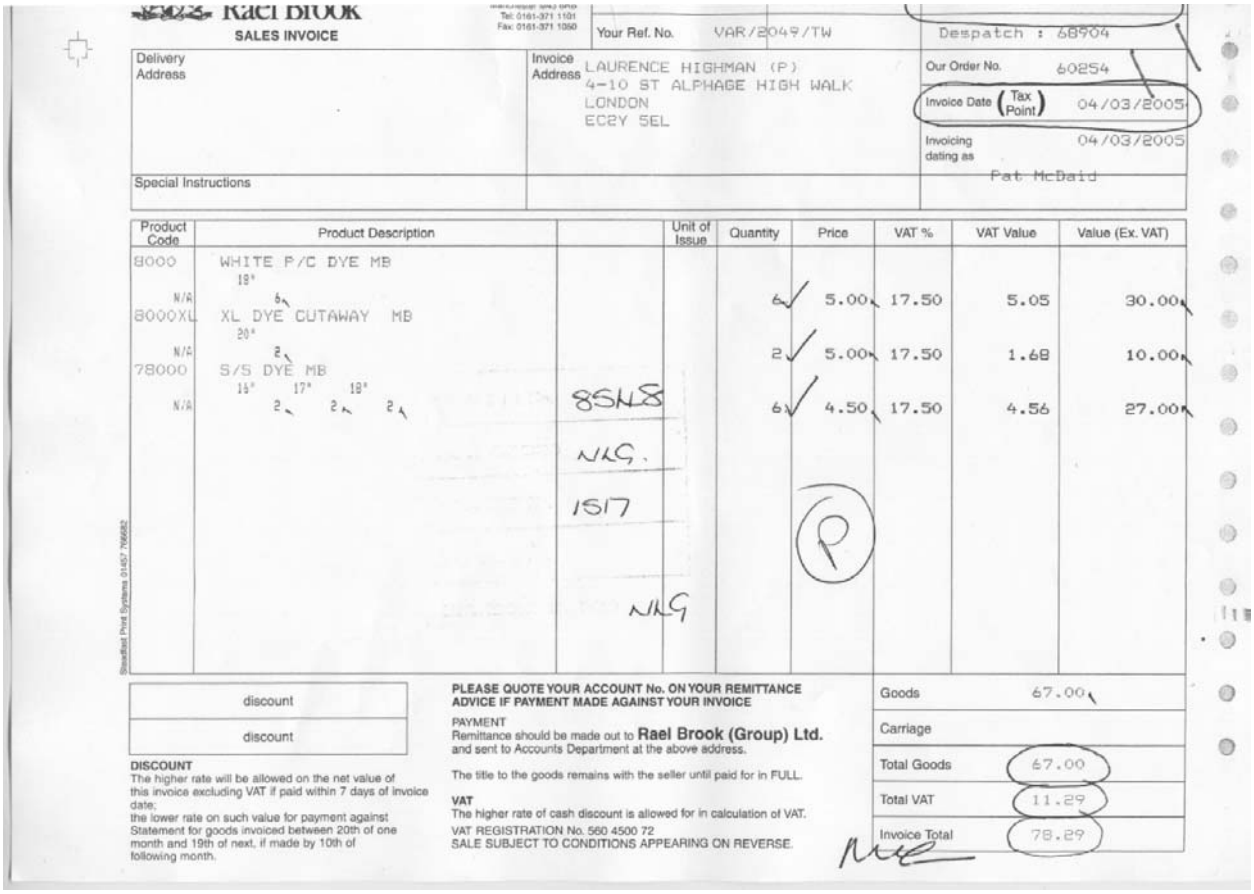
Signature

Appendix D2 – Sample order book excerpt for 'Rael Brook'



7/3	8548	Rael Brook 6x 8000 L/S White 18 2x " " " 20 2x 78000 S/S " 16 2x " " " 17 2x " " " 18	68904	2049	ON/3/05 Invoice 69104
7/3	8549	DAVern 6x 01149 Black Fleece XL	0516906	2060	ON/3/05 Invoice ON81133
7/3	8550	Disley 3x C896/L White L/S Shirts 20"	53000	2053	03/03/05 Invoice 90309
7/3	8551	Disley 2x C896/L White L/S Shirts 20"	52999	2035	03/03/05 Invoice 90309
7/3	8552	UKD 1x MA36A 10 1x MA22A 10 1x MA22A 11 1x M586A 9 1x M586A 11 1x M704A 8 1x M694A 11 2x M775A 11 1x M775A 10 2x M775A 8	532282	2047	09/3/05 Invoice 532239
7/3	8553	Loakes 1x Rome Black 9½ 1x " Burgundy 9½	D126575	2006	ON/3/05 Invoice 228189

Appendix D3 – Sales Invoice from 'Rael Brook'



Rael Brook
SALES INVOICE

Tel: 0161-371 1101
Fax: 0161-371 1050

Your Ref. No. VAR/2049/TW

Despatch : 68904

Delivery Address

Invoice Address
LAURENCE HIGHMAN (P)
4-10 ST ALPHAGE HIGH WALK
LONDON
EC2Y 5EL

Our Order No. 60254

Invoice Date (Tax Point) 04/03/2005

Invoicing dating as 04/03/2005

Special Instructions

Pat McDaid

Product Code	Product Description		Unit of Issue	Quantity	Price	VAT %	VAT Value	Value (Ex. VAT)
8000	WHITE P/C DYE MB							
	18"							
N/A		6		6	5.00	17.50	5.05	30.00
8000XL	XL DYE CUTAWAY MB							
	20"							
N/A		2		2	5.00	17.50	1.68	10.00
78000	S/S DYE MB							
	16" 17" 18"							
N/A	2 2 2			6	4.50	17.50	4.56	27.00

8SNS
NKG.
1517
NKG

discount	
discount	

DISCOUNT
The higher rate will be allowed on the net value of this invoice excluding VAT if paid within 7 days of invoice date:
the lower rate on such value for payment against Statement for goods invoiced between 20th of one month and 19th of next, if made by 10th of following month.

PLEASE QUOTE YOUR ACCOUNT No. ON YOUR REMITTANCE ADVICE IF PAYMENT MADE AGAINST YOUR INVOICE

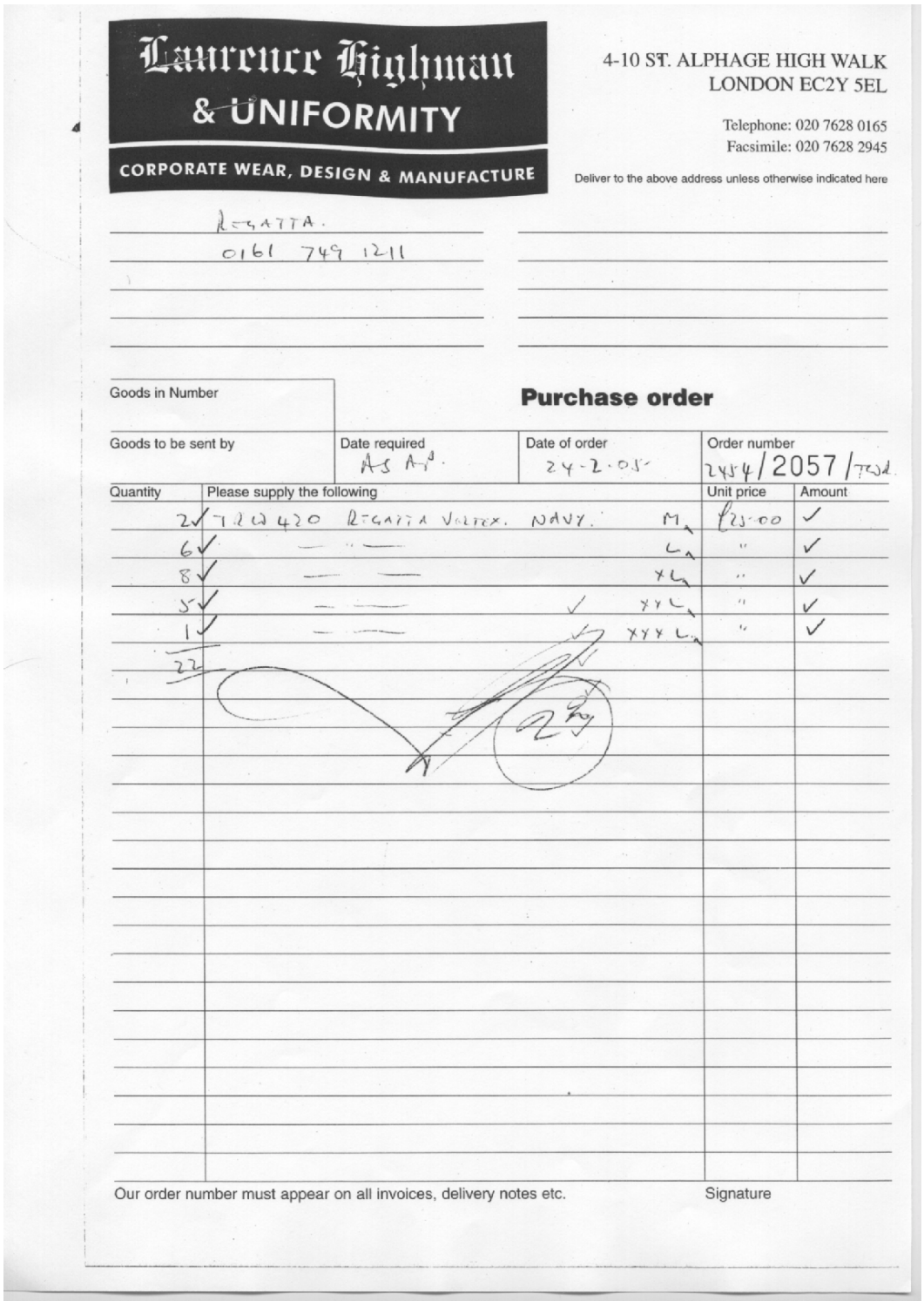
PAYMENT
Remittance should be made out to **Rael Brook (Group) Ltd.** and sent to Accounts Department at the above address.

The title to the goods remains with the seller until paid for in FULL.

VAT
The higher rate of cash discount is allowed for in calculation of VAT.
VAT REGISTRATION No. 560 4500 72
SALE SUBJECT TO CONDITIONS APPEARING ON REVERSE.

Goods	67.00
Carriage	
Total Goods	67.00
Total VAT	11.29
Invoice Total	78.29

Appendix D4 – Sample Purchase order for 'Regatta'



Laurence Eyhman & UNIFORMITY

CORPORATE WEAR, DESIGN & MANUFACTURE

4-10 ST. ALPHAGE HIGH WALK
LONDON EC2Y 5EL

Telephone: 020 7628 0165
Facsimile: 020 7628 2945

Deliver to the above address unless otherwise indicated here

REGATTA.
0161 749 1211

Goods in Number

Purchase order

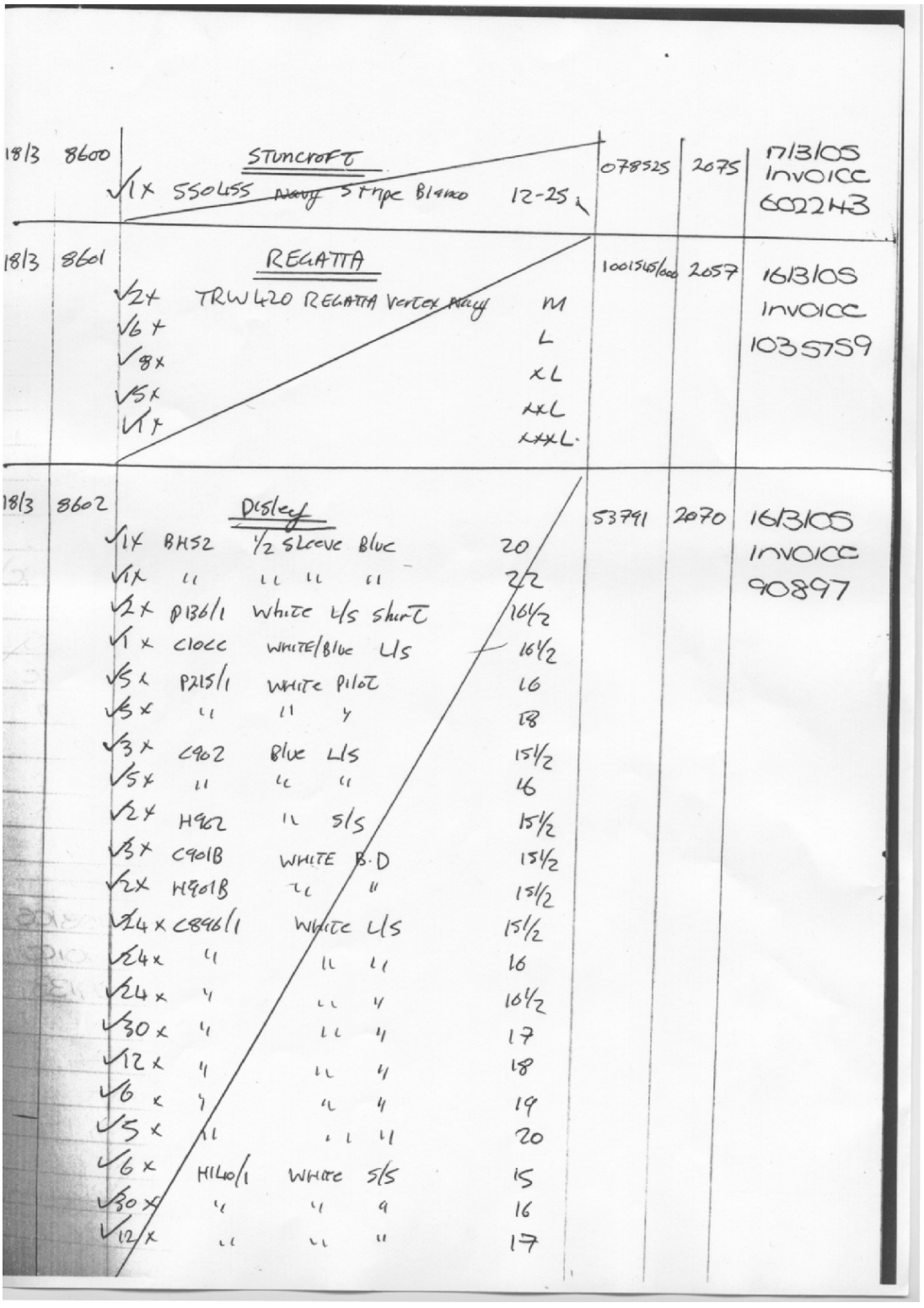
Goods to be sent by	Date required	Date of order	Order number
	AS A.P.	24-2-05	2454/2057/TSH

Quantity	Please supply the following		Unit price	Amount
2 ✓	TRA420 REGATTA VALTEX. NAVY.	M	£25.00	✓
6 ✓	— " —	L	"	✓
8 ✓	— —	XL	"	✓
5 ✓	— — ✓	XXL	"	✓
1 ✓	— — ✓	XXXL	"	✓
22				

Our order number must appear on all invoices, delivery notes etc.

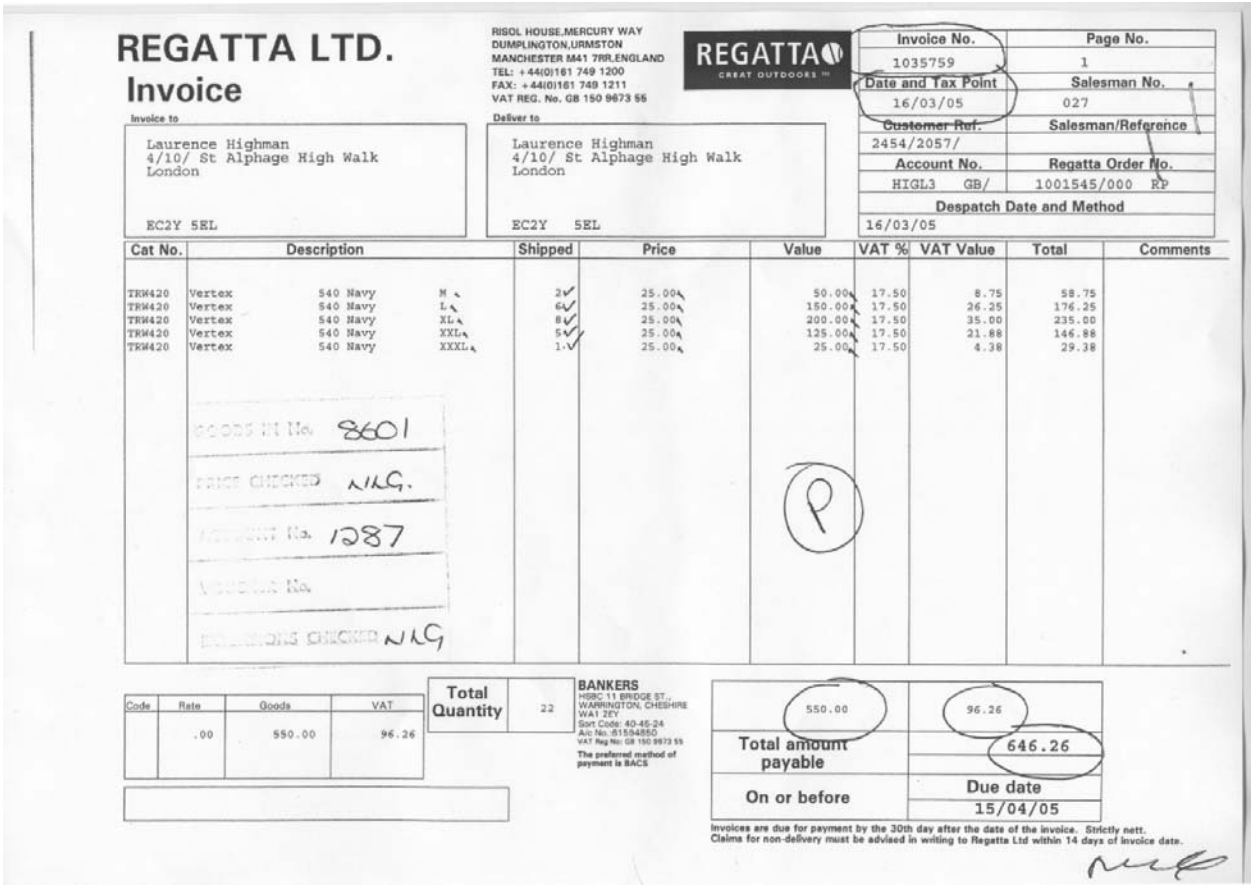
Signature

Appendix D5 – Sample order book excerpt for 'Regatta'



18/3	8600	STUNCROFT		078525	2075	17/3/05 Invoice 602143
		✓1 x 550455 Navy Stripe Blanco	12-25			
18/3	8601	REGATTA		100156/1000	2057	16/3/05 Invoice 1035759
		✓2 x TRW420 REGATTA Vertex Polo	M			
		✓6 x	L			
		✓8 x	XL			
		✓5 x	XXL			
		✓1 x	XXXL			
18/3	8602	Disley		53791	2070	16/3/05 Invoice 90897
		✓1 x BH52 ½ Sleeve Blue	20			
		✓1 x " " " "	22			
		✓2 x P136/1 White L/S shirt	16½			
		✓1 x C10CC White/Blue L/S	16½			
		✓5 x P215/1 White Pilot	16			
		✓5 x " " "	18			
		✓3 x C902 Blue L/S	15½			
		✓5 x " " "	16			
		✓2 x H902 " S/S	15½			
		✓3 x C901B White B.D	15½			
		✓2 x H901B " "	15½			
		✓4 x C896/1 White L/S	15½			
		✓4 x " " "	16			
		✓4 x " " "	16½			
		✓30 x " " "	17			
		✓12 x " " "	18			
		✓6 x " " "	19			
		✓5 x " " "	20			
		✓6 x H140/1 White S/S	15			
		✓30 x " " "	16			
		✓12 x " " "	17			

Appendix D6 – Sales Invoice from 'Regatta'



REGATTA LTD.

Invoice

RISOL HOUSE,MERCURY WAY
DUMPLINGTON,URMSTON
MANCHESTER M41 7RR,ENGLAND
TEL: +44(0)161 749 1200
FAX: +44(0)161 749 1211
VAT REG. No. GB 150 9673 55

REGATTA
GREAT OUTDOORS

Invoice No.	Page No.
1035759	1
Date and Tax Point	Salesman No.
16/03/05	027
Customer Ref.	Salesman/Reference
2454/2057/	
Account No.	Regatta Order No.
HIGL3 GB/	1001545/000 RP
Despatch Date and Method	
16/03/05	

Invoice to

Laurence Highman
4/10/ St Alphage High Walk
London

EC2Y 5EL

Deliver to

Laurence Highman
4/10/ St Alphage High Walk
London

EC2Y 5EL

Cat No.	Description			Shipped	Price	Value	VAT %	VAT Value	Total	Comments
TRW420	Vertex	540 Navy	M	2	25.00	50.00	17.50	8.75	58.75	
TRW420	Vertex	540 Navy	L	6	25.00	150.00	17.50	26.25	176.25	
TRW420	Vertex	540 Navy	XL	8	25.00	200.00	17.50	35.00	235.00	
TRW420	Vertex	540 Navy	XXL	5	25.00	125.00	17.50	21.88	146.88	
TRW420	Vertex	540 Navy	XXXL	1	25.00	25.00	17.50	4.38	29.38	

GOODS IN No. 8601
PRICE CHECKED NLG.
AMOUNT No. 1287
VOUCHER No.
EXTENSIONS CHECKED NLG

Code	Rate	Goods	VAT
	.00	550.00	96.26

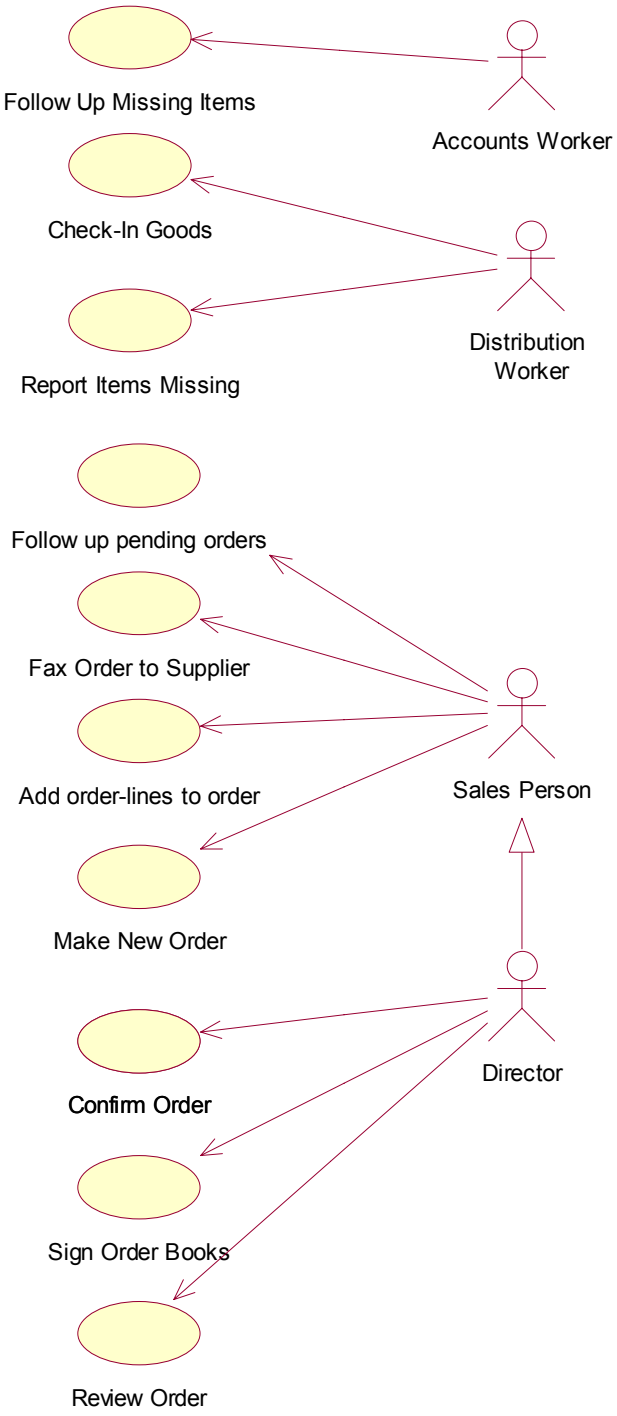
Total Quantity 22

BANKERS
HSBC 11 BRIDGE ST., WARRINGTON, CHESHIRE WA1 2EY
Sort Code: 40-45-24
A/c No. 61594850
VAT Reg No: GB 150 9673 55
The preferred method of payment is BACS

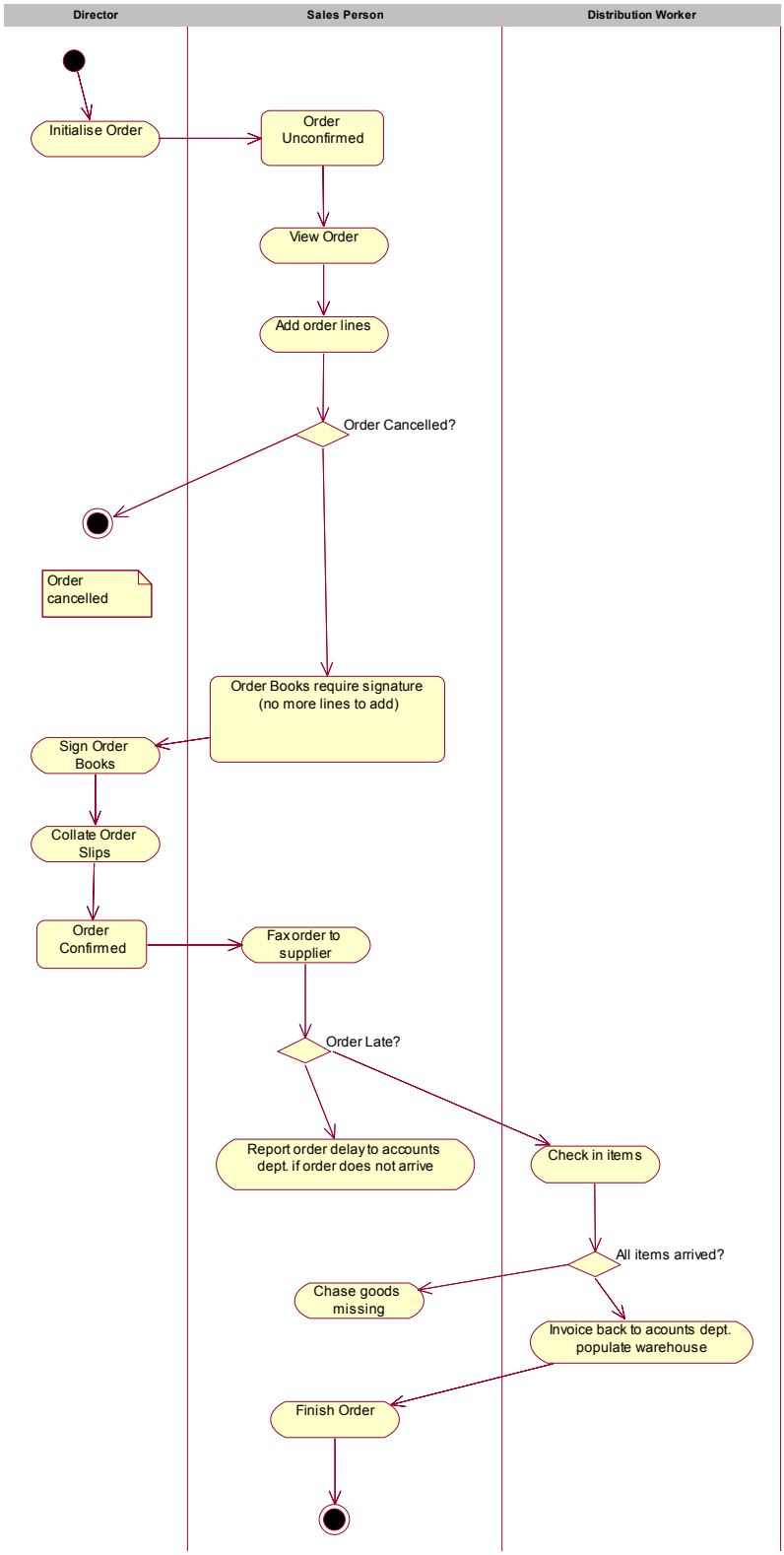
550.00	96.26
Total amount payable	646.26
On or before	Due date 15/04/05

Invoices are due for payment by the 30th day after the date of the invoice. Strictly nett.
Claims for non-delivery must be advised in writing to Regatta Ltd within 14 days of invoice date.

Appendix E – Use Case Diagram



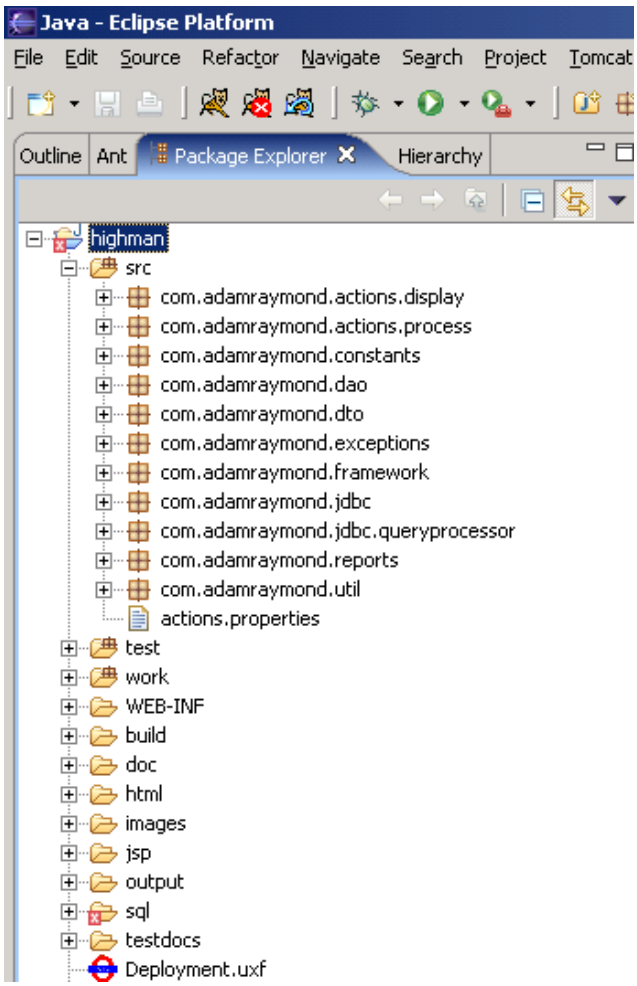
Appendix F – Activity Diagram



Appendix G – Package Structure of System

Screen Shot depicts overall package structure for the finished system.

com.adamraymond.actions.display – All display Action Java Files
com.adamraymond.actions.process – All process Action Java Files
com.adamraymond.constants – System constants such as user access rights variables
com.adamraymond.dao– All Data Access Objects
com.adamraymond.dto– All Data Transfer Objects
com.adamraymond.exceptions – Classes to build collection of errors to display in JSP page
com.adamraymond.framework – Based on Geary Framework with enhancements.
com.adamraymond.jdbc – Classes that make up JDBC layer of system
com.adamraymond.jdbc.queryprocessor – Query Processors for JBDC layer
com.adamraymond.reports – Classes that build iText based PDF documents.



Appendix H – E-R Diagram for Proposed Solution

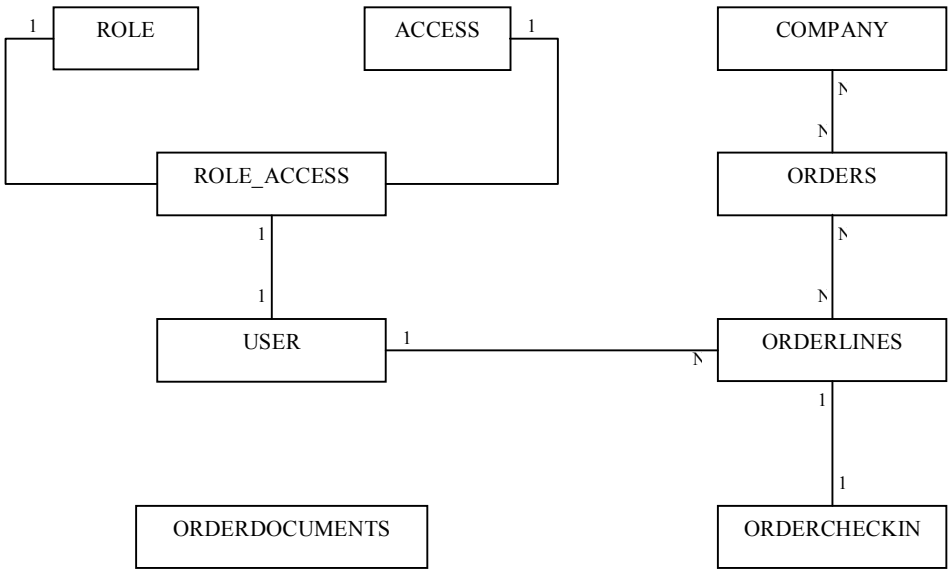


Table Descriptions

ROLE: Specifies the 3 types of access available to the system

ACCESS: Specifies all the possible actions that can be performed in the system

ROLE_ACCESS: Specifies all possible actions that can occur for each access type

USER: Contains user credentials and specifies access type user is assigned to.

ORDERS: Contains order information such as date required and order total

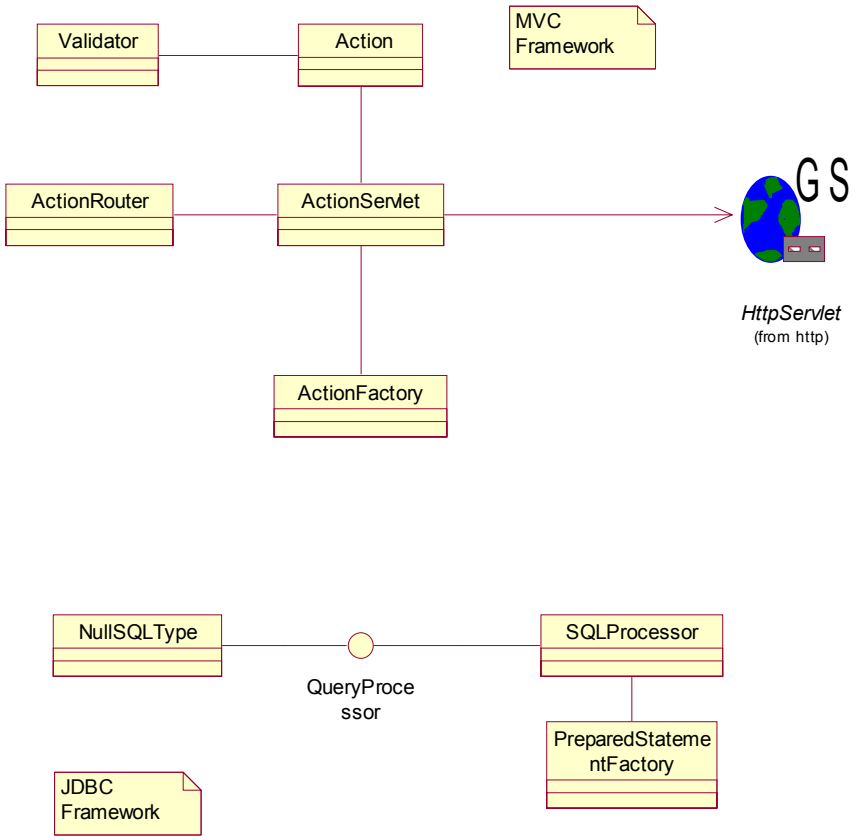
ORDERLINES: contains order-lines, each order-line being referenced to an order record.

COMPANY: Contains company credentials such as address and telephone number

ORDERCHECKIN: specifies quantities received for an order-line (relating to an order)

ORDERDOCUMENTS: Archives all finished orders as PDF file, permanently stored on disk

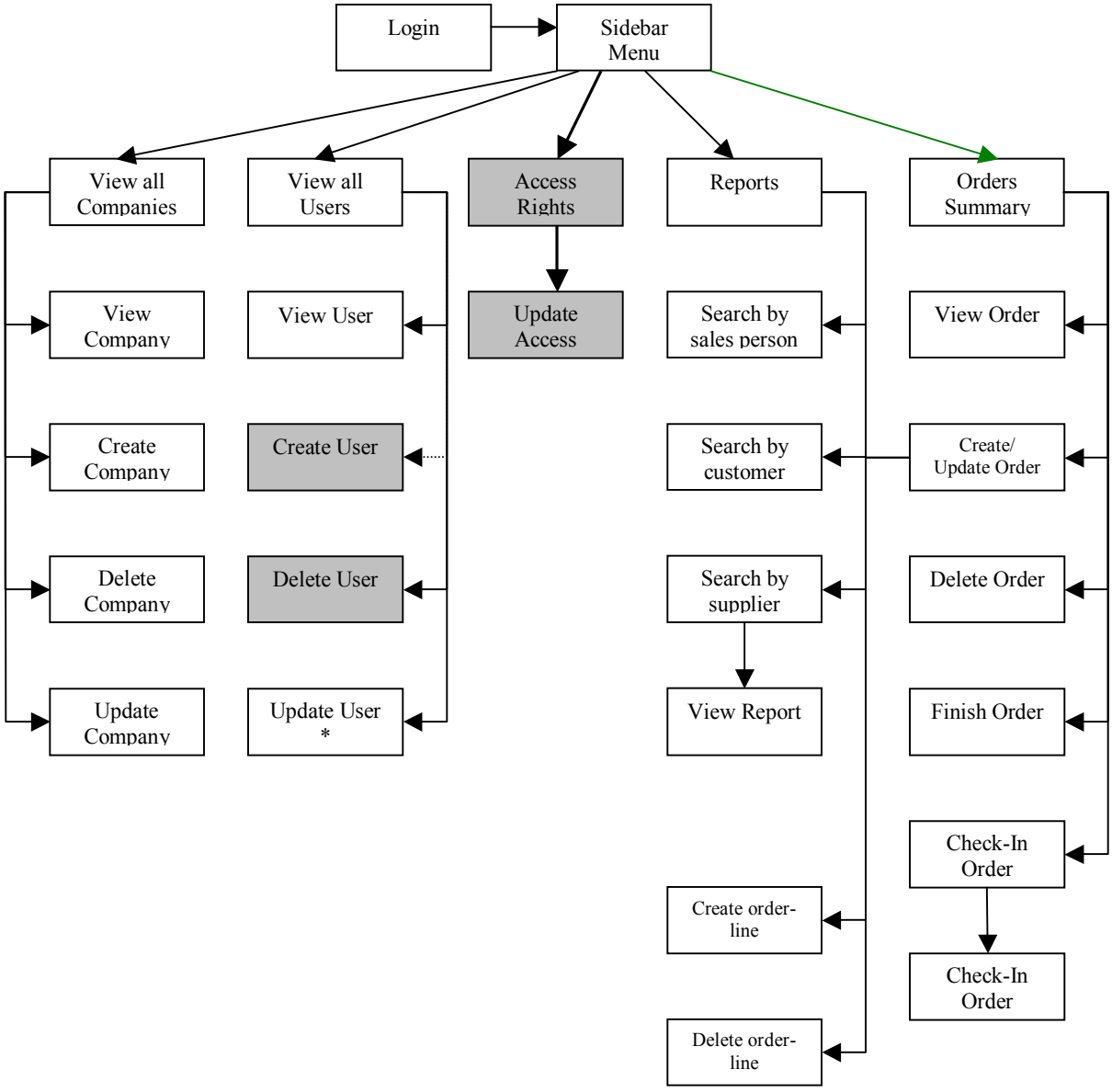
Appendix I – System Concept Level Class Diagram



PHASES OF IMPLEMENTATION

Phase No.	Object Implemented
1	Companies
2	Users, Access Rights
3	Orders
4	Order lines, PDF writer
5	Order Check-In
6	Order reports

Appendix J – Navigation Map for Proposed System



* Please note that the Update User command is available to all users, however only administrator will be able to see the option to update a users access rights.

Appendix K – Database Schema

Table Name	Table Schema
Access	access (id, code, description)
Company	company (id, name, addressline1, addressline2, addressline3, town, county, postcode, telephone, fax, deleted)
Ordercheckin	ordercheckin (*order_id, orderline_id*, qty_received, comment)
Orderdocument	orderdocument (id, order_no, doc, date)
Orderline	orderline (id, quantity, code, size, colour, description, unitprice, total, *order_id, user_id, comp_id*)
Orders	orders (id, code, delivery_method, date_required, date_ordered, details, total, *confirm_user_id, comp_id,* deleted)
Role	role (id, name)
Role_access	role_access (*role_id, access_id*)
User	user (id, ***username***, fullname, ***reference***, extension, password, *accessrights*, deleted)

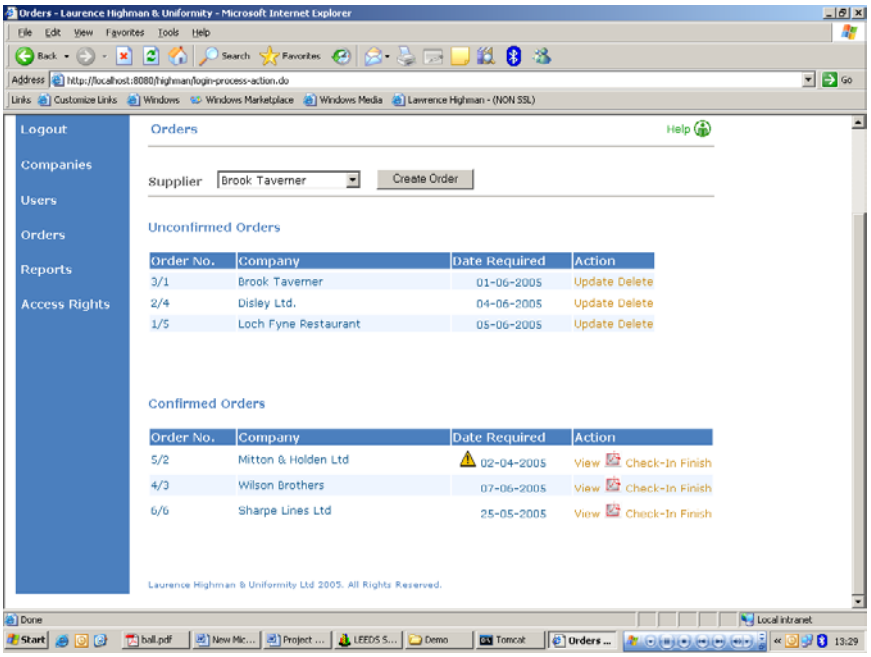
LEGEND:

Primary keys are underlined

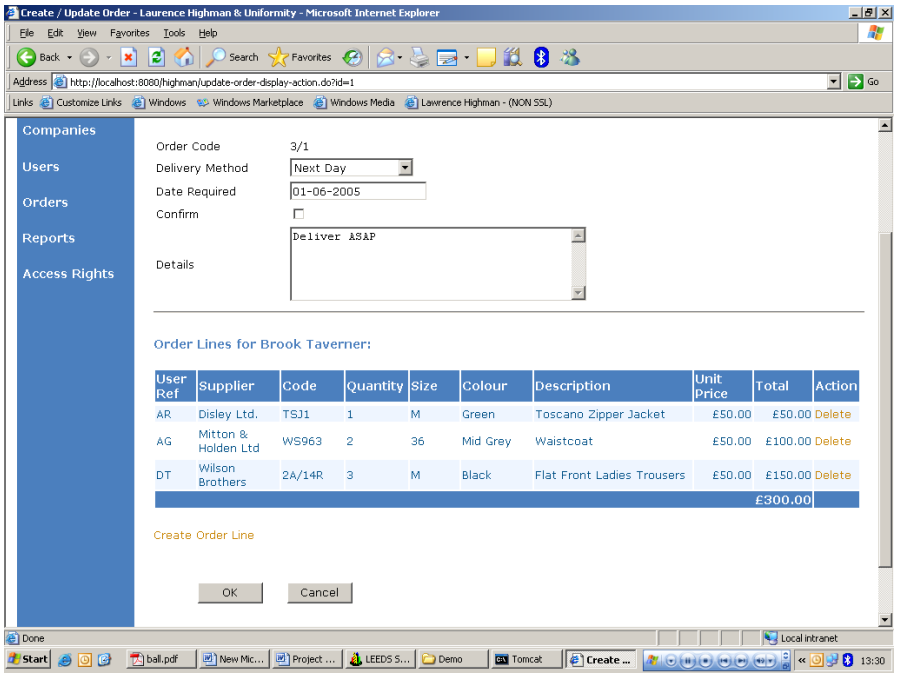
Foreign Keys are italicised

Unique keys are both italicised and emboldened

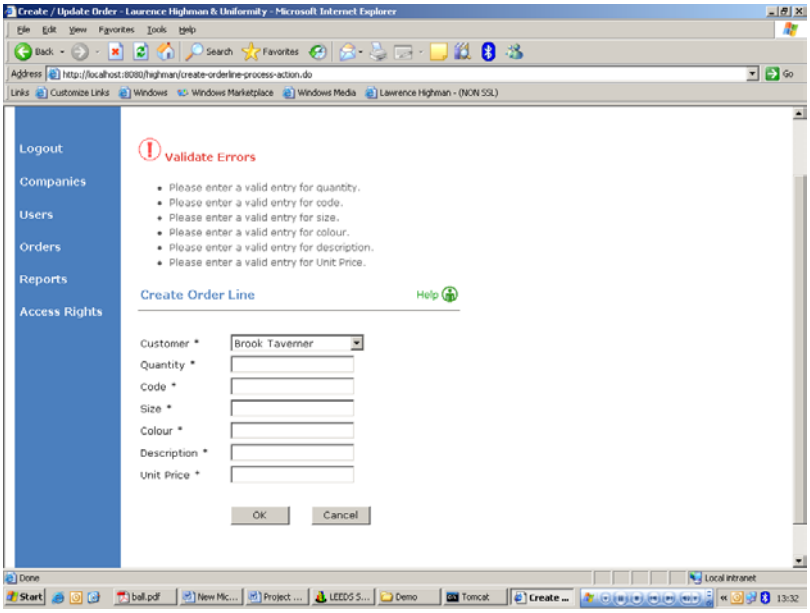
Appendix L – System Screen Shots


Orders - Laurence Highman & Uniformity - Microsoft Internet Explorer
Logout
Companies
Users
Orders
Reports
Access Rights
Orders
Help
Supplier
Brook Taverner
Create Order
Unconfirmed Orders
Order No. | Company | Date Required | Action
3/1 | Brook Taverner | 01-06-2005 | Update Delete
2/4 | Disley Ltd. | 04-06-2005 | Update Delete
1/5 | Loch Fyne Restaurant | 05-06-2005 | Update Delete
Confirmed Orders
Order No. | Company | Date Required | Action
5/2 | Mitton & Holden Ltd | 02-04-2005 | View Check-In Finish
4/3 | Wilson Brothers | 07-06-2005 | View Check-In Finish
6/6 | Sharpe Lines Ltd | 25-05-2005 | View Check-In Finish
Laurence Highman & Uniformity Ltd 2005. All Rights Reserved.

Screen shot showing confirmed and unconfirmed orders in the correct table view. Orders which are overdue are shown with a ‘hazard’ symbol

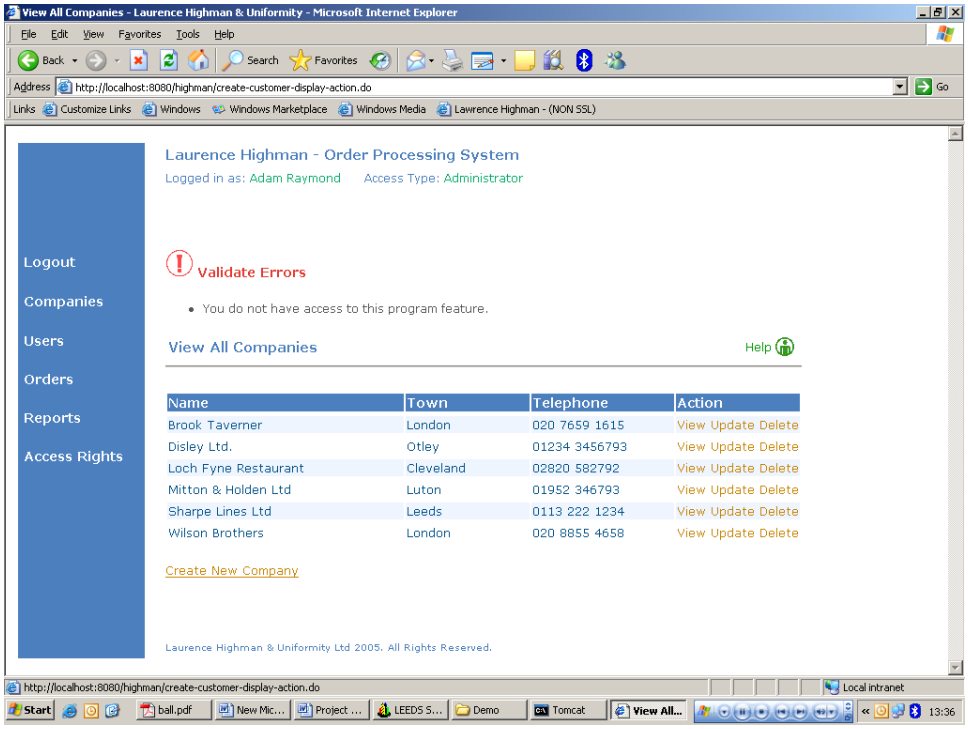

Create / Update Order - Laurence Highman & Uniformity - Microsoft Internet Explorer
Companies
Users
Orders
Reports
Access Rights
Order Code 3/1
Delivery Method Next Day
Date Required 01-06-2005
Confirm
Details Deliver ASAP
Order Lines for Brook Taverner:
User Ref | Supplier | Code | Quantity | Size | Colour | Description | Unit Price | Total | Action
AR | Disley Ltd. | TSJ1 | 1 | M | Green | Toscano Zipper Jacket | £50.00 | £50.00 | Delete
AG | Mitton & Holden Ltd | WS963 | 2 | 36 | Mid Grey | Waistcoat | £50.00 | £100.00 | Delete
DT | Wilson Brothers | 2A/14R | 3 | M | Black | Flat Front Ladies Trousers | £50.00 | £150.00 | Delete
£300.00
Create Order Line
OK
Cancel

Screen shot showing an order with its associated order-lines. The order total is calculated automatically when order-lines are added or deleted.

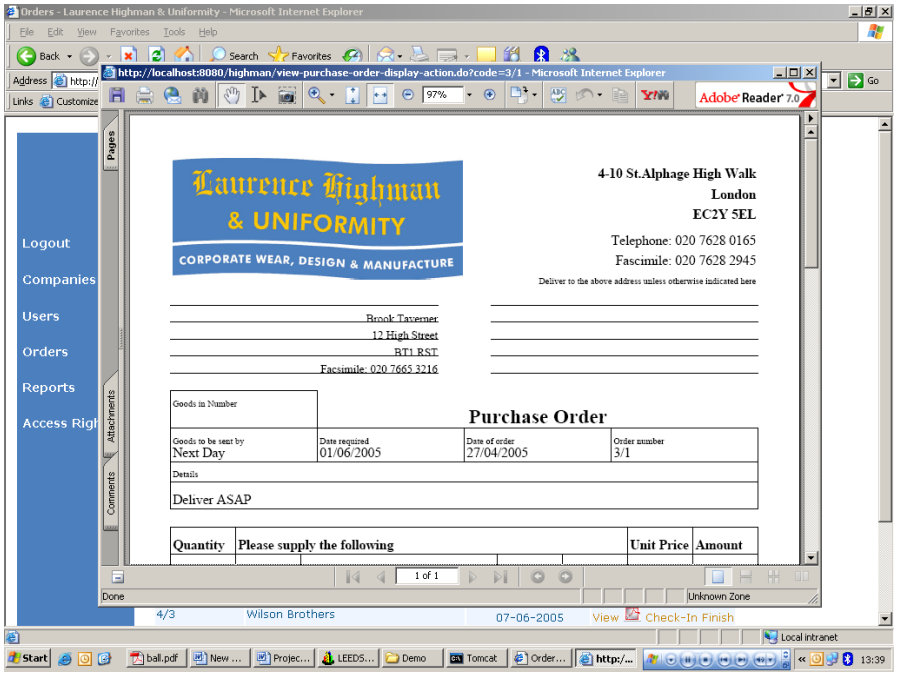


Screen showing collection of errors returned from validation exceptions thrown in the controller pattern. These errors have been assigned a customised error message to show the user which field must be validated before proceeding.

Screen shot showing CSS scriptlet application. System paints the next line in the list the opposite of the previous entry. This results in easier to read two tone tables.



Screen shot showing the result of the determineAccess method, as defined in the Implementation chapter, when access is denied to a particular action.



Screen shot showing PDF generated from database contents for the appropriate order.

Appendix M – Unit Testing Results

NB: Where INITIAL FAILURE is mentioned for a test success, it refers to a detected error. The error was resolved and the test was then successful. The course of actions to resolve the errors are given.

Company Object

View All Companies				
Test Case	Test	Expected Result	Actual Result	Success? (Comment for changes)
Sidebar routes to view all companies page.	Select Companies from sidebar menu.	List of the Name, Town, Telephone fields for all companies in companies table, in alphabetical order.	All companies in companies table displayed in alphabetical order, showing Name, Town and Telephone fields.	Test Successful.
Links displayed correctly.	Visual verification of links existence.	Every company entry on page has a View, Update and Delete Action. Create link also displayed under table.	All entries are displayed with their own View, Update and Delete action links. Create link also displayed.	Test Successful.
Help icon displayed correctly and routes to correct help page.	Click on Help link on page.	Link displayed and opens pop-up window for relevant help page.	Link displayed and opens pop-up window for relevant help page.	Test Successful.

View Company				
Test Case	Test	Expected Result	Actual Result	Success/ Failure? (Comment for changes)
View action routes to view company page.	Select the View action for a company record.	System routes to view company page.	System routed to view company page.	Test Successful.
Page displays Name caption.	Attempt to edit caption.	Name caption displayed, non- editable.	Name caption displayed and non-editable.	Test Successful.
Page displays Address Line 1 caption.	Attempt to edit caption.	Address Line 1 caption displayed, non- editable.	Address Line 1 caption displayed and non-editable.	Test Successful.
Page displays Address Line 2 caption.	Attempt to edit caption.	Address Line 2 caption displayed, non- editable.	Address Line 2 caption displayed and non- editable.	Test Successful.
Page displays Address Line 3 caption.	Attempt to edit 7caption.	Address Line 3 caption displayed, non- editable.	Address Line 3 caption displayed and non- editable.	Test Successful.
Page displays Town caption.	Attempt to edit caption.	Town caption displayed, non- editable.	Town caption displayed and non- editable.	Test Successful.
Page displays County caption.	Attempt to edit caption.	County caption displayed, non- editable.	County caption displayed and non- editable.	Test Successful.
Page displays Postcode caption.	Attempt to edit caption.	Postcode caption displayed, non- editable.	Postcode caption displayed, non- editable.	Test Successful.
Page displays Telephone caption.	Attempt to edit caption.	Telephone caption displayed, non- editable.	Telephone caption displayed, non- editable.	Test Successful.
Page displays Fax caption.	Attempt to edit caption.	Fax caption displayed, non-editable.	Fax caption displayed, non-editable.	Test Successful.
Page displays Address Line 1 field.	Attempt to edit field	Address Line 1 field displayed, non- editable.	Address Line 1 field displayed, non-editable.	Test Successful.
Page displays Address Line 2 field.	Attempt to edit field	Address Line 2 field displayed, non- editable.	Address Line 2 field displayed, non- editable.	Test Successful.
Page displays Address Line 3 field.	Attempt to edit field	Address Line 3 field displayed, non- editable.	Address Line 3 field displayed, non- editable.	Test Successful.
Page displays Town field.	Attempt to edit field	Town field displayed, non-editable.	Town field displayed, non-editable.	Test Successful.
Page displays County field.	Attempt to edit field	County field displayed, non-editable.	County field displayed, non-editable.	Test Successful.
Page displays Postcode field.	Attempt to edit field	Postcode field displayed, non- editable.	Postcode field displayed, non-editable.	Test Successful.
Page displays Telephone field.	Attempt to edit field	Telephone field displayed, non- editable.	Telephone field displayed, non- editable.	Test Successful.
Page displays Fax field.	Attempt to edit field	Fax field displayed, non-editable.	Fax field displayed, non-editable.	Test Successful.

OK Button displayed and routes back to view all companies page.	Click on OK button.	Button displayed and routes back to view all companies page.	Button displayed and routes back to view all companies page.	Test Successful. INITIAL FAILURE corrected when routed to index page.
Help icon displayed correctly and routes to correct help page.	Click on Help link on page.	Link displayed and opens pop-up window for relevant help page.	Link displayed and opens pop-up window for relevant help page.	Test Successful.

Create Company				
Test Case	Test	Expected Result	Actual Result	Success/ Failure? (Comment for changes)
Create New Company action routes to view company page.	Select the Create New Company action.	System routes to create company page.	System routed to create company page.	Test Successful.
Page displays Name caption.	Attempt to edit caption.	Name caption displayed, non- editable.	Name caption displayed, non-editable.	Test Successful.
Page displays Address Line 1 caption.	Attempt to edit caption.	Address Line 1 caption displayed, non- editable.	Address Line 1 caption displayed, non-editable.	Test Successful.
Page displays Address Line 2 caption.	Attempt to edit caption.	Address Line 2 caption displayed, non- editable.	Address Line 2 caption displayed, non- editable.	Test Successful.
Page displays Address Line 3 caption.	Attempt to edit caption.	Address Line 3 caption displayed, non- editable.	Address Line 3 caption displayed, non- editable.	Test Successful.
Page displays Town caption.	Attempt to edit caption.	Town caption displayed, non- editable.	Town caption displayed, non-editable.	Test Successful.
Page displays County caption.	Attempt to edit caption.	County caption displayed, non- editable.	County caption displayed, non- editable.	Test Successful.
Page displays Postcode caption.	Attempt to edit caption.	Postcode caption displayed, non- editable.	Postcode caption displayed, non- editable.	Test Successful.
Page displays Telephone caption.	Attempt to edit caption.	Telephone caption displayed, non- editable.	Telephone caption displayed, non- editable.	Test Successful.
Page displays Fax caption.	Attempt to edit caption.	Fax caption displayed and non- editable.	Fax caption displayed, non-editable.	Test Successful.
Page displays Name text field.	Enter text into field.	Name text field accepts text up to 45 characters.	Name text field accepts text up to 45 characters.	Test Successful.
Page displays Address Line 1 text field.	Enter text into field.	Address Line 1 text field accepts text up to 45 characters.	Address Line 1 text field accepts text up to 45 characters.	Test Successful.
Page displays Address Line 2 text field.	Enter text into field.	Address Line 2 text field accepts text up to 45 characters.	Address Line 2 text field accepts text up to 45 characters.	Test Successful.
Page displays Address Line 3 text field.	Enter text into field.	Address Line 2 text field accepts text up to 45 characters.	Address Line 2 text field accepts text up to 45 characters.	Test Successful.
Page displays Town text field.	Enter text into field.	Town text field accepts text up to 45 characters.	Town text field accepts text up to 45 characters.	Test Successful.
Page displays County text field.	Enter text into field.	County text field accepts text up to 45 characters.	County text field accepts text up to 45 characters.	Test Successful.
Page displays Postcode text field.	Enter text into field.	Postcode text field accepts text up to 45 characters.	Postcode text field accepts text up to 45 characters.	Test Successful.
Page displays Telephone text field.	Enter text into field.	Telephone text field accepts text up to 15 characters.	Telephone text field accepts text up to 15 characters.	Test Successful. INITIAL FAILURE corrected from 45 to 15 characters.
Page displays Fax text field.	Enter text into field.	Fax text field accepts text up to 15 characters.	Fax text field accepts text up to 15 characters.	Test Successful. INITIAL FAILURE corrected from 45 to 15 characters.
Page displays Name validation error message.	Attempt to submit null field.	Page displays error message for Name Field on submit attempt.	Page displays error message for Name Field on submit attempt.	Test Successful.

Page displays Address Line 1 validation error message.	Attempt to submit null field.	Page displays error message for Address Line 1 Field on submit attempt.	Page displays error message for Address Line 1 Field on submit attempt.	Test Successful.
Page displays Town validation error message.	Attempt to submit null field.	Page displays error message for Town Field on submit attempt.	Page displays error message for Town Field on submit attempt.	Test Successful.
Page displays County validation error message.	Attempt to submit null field.	Page displays error message for County Field on submit attempt.	Page displays error message for County Field on submit attempt.	Test Successful.
Page displays Postcode validation error message.	Attempt to submit null field.	Page displays error message for Postcode Field on submit attempt.	Page displays error message for Postcode Field on submit attempt.	Test Successful.
Page displays Telephone validation error message.	Attempt to submit null field.	Page displays error message for Telephone Field on submit attempt.	Page displays error message for Telephone Field on submit attempt.	Test Successful.
Page displays Fax validation error message.	Attempt to submit null field.	Page displays error message for Fax Field on submit attempt.	Page displays error message for Fax Field on submit attempt.	Test Successful.
OK Button displayed and routes back to view all companies page. Creates new entry.	Click on OK button.	Button displayed and routes back to view all companies page. New entry created.	Button displayed and routes back to view all companies page. New entry created.	Test Successful.
Cancel Button displayed and routes back to view all companies page. Does not create new entry.	Click on Cancel button.	Button displayed and routes back to view all companies page. No new entry created.	Button displayed and routes back to view all companies page. No new entry created.	Test Successful.
Help icon displayed correctly and routes to correct help page.	Click on Help link on page.	Link displayed and opens pop-up window for relevant help page.	Link displayed and opens pop-up window for relevant help page.	Test Successful.

Update Company				
Test Case	Test	Expected Result	Actual Result	Success/ Failure? (Comment for changes)
Update action routes to update company page.	Select the Update action for a company record.	System routes to update company page.	System routed to update company page.	Test Successful.
Page displays Name caption.	Attempt to edit caption.	Name caption displayed, non- editable.	Name caption displayed, non-editable.	Test Successful.
Page displays Address Line 1 caption.	Attempt to edit caption.	Address Line 1 caption displayed, non- editable.	Address Line 1 caption displayed, non-editable.	Test Successful.
Page displays Address Line 2 caption.	Attempt to edit caption.	Address Line 2 caption displayed, non- editable.	Address Line 2 caption displayed, non- editable.	Test Successful.
Page displays Address Line 3 caption.	Attempt to edit caption.	Address Line 3 caption displayed, non- editable.	Address Line 3 caption displayed, non- editable.	Test Successful.
Page displays Town caption.	Attempt to edit caption.	Town caption displayed, non- editable.	Town caption displayed, non-editable.	Test Successful.
Page displays County caption.	Attempt to edit caption.	County caption displayed, non- editable.	County caption displayed and non- editable.	Test Successful.
Page displays Postcode caption.	Attempt to edit caption.	Postcode caption displayed, non- editable.	Postcode caption displayed, non- editable.	Test Successful.
Page displays Telephone caption.	Attempt to edit caption.	Telephone caption displayed, non- editable.	Telephone caption displayed, non- editable.	Test Successful.
Page displays Fax caption.	Attempt to edit caption.	Fax caption displayed, non-editable.	Fax caption displayed and, -editable.	Test Successful.
Page displays Name text field contents from database.	Update text field.	Displays field contents from database. Name text field accepts text up to 45 characters.	Displays field contents from database. Name text field accepts text up to 45 characters.	Test Successful.
Page displays Address Line 1 text field contents from database.	Update text field.	Displays field contents from database. Address Line 1 text field accepts text up to	Displays field contents from database. Address Line 1 text field accepts text up to 45	Test Successful.

		45 characters.	characters.	
Page displays Address Line 2 text field contents from database.	Update text field.	Displays field contents from database. Address Line 2 text field accepts text up to 45 characters.	Displays field contents from database. Address Line 2 text field accepts text up to 45 characters.	Test Successful.
Page displays Address Line 3 text field contents from database.	Update text field.	Displays field contents from database. Address Line 2 text field accepts text up to 45 characters.	Displays field contents from database. Address Line 2 text field accepts text up to 45 characters.	Test Successful.
Page displays Town text field contents from database.	Update text field.	Displays field contents from database. Town text field accepts text up to 45 characters.	Displays field contents from database. Town text field accepts text up to 45 characters.	Test Successful.
Page displays County text field contents from database.	Update text field.	Displays field contents from database. County text field accepts text up to 45 characters.	Displays field contents from database. County text field accepts text up to 45 characters.	Test Successful.
Page displays Postcode text field contents from database.	Update text field.	Displays field contents from database. Postcode text field accepts text up to 45 characters.	Displays field contents from database. Postcode text field accepts text up to 45 characters.	Test Successful.
Page displays Telephone text field contents from database.	Update text field.	Displays field contents from database. Telephone text field accepts text up to 15 characters.	Displays field contents from database. Telephone text field accepts text up to 15 characters.	Test Successful.
Page displays Fax text field contents from database.	Update text field.	Displays field contents from database. Fax text field accepts text up to 15 characters.	Displays field contents from database. Fax text field accepts text up to 15 characters.	Test Successful.
Page displays Name validation error message.	Attempt to submit null field.	Page displays error message for Name Field on submit attempt.	Page displays error message for Name Field on submit attempt.	Test Successful.
Page displays Address Line 1 validation error message.	Attempt to submit null field.	Page displays error message for Address Line 1 Field on submit attempt.	Page displays error message for Address Line 1 Field on submit attempt.	Test Successful.
Page displays Town validation error message.	Attempt to submit null field.	Page displays error message for Town Field on submit attempt.	Page displays error message for Town Field on submit attempt.	Test Successful.
Page displays County validation error message.	Attempt to submit null field.	Page displays error message for County Field on submit attempt.	Page displays error message for County Field on submit attempt.	Test Successful.
Page displays Postcode validation error message.	Attempt to submit null field.	Page displays error message for Postcode Field on submit attempt.	Page displays error message for Postcode Field on submit attempt.	Test Successful.
Page displays Telephone validation error message.	Attempt to submit null field.	Page displays error message for Telephone Field on submit attempt.	Page displays error message for Telephone Field on submit attempt.	Test Successful.
Page displays Fax validation error message.	Attempt to submit null field.	Page displays error message for Fax Field on submit attempt.	Page displays error message for Fax Field on submit attempt.	Test Successful.
OK Button displayed and routes back to view all companies page where updated entry shown in table.	Click on OK button.	Button displayed and routes back to view all companies page. Entry updated.	Button displayed and routes back to view all companies page. Entry updated.	Test Successful.
Cancel Button displayed and routes back to view all companies page. Does not update entry.	Click on Cancel button.	Button displayed and routes back to view all companies page. No update.	Button displayed and routes back to view all companies page. No update.	Test Successful.
Help icon displayed correctly and routes to correct help page.	Click on Help link on page.	Link displayed and opens pop-up window for relevant help page.	Link displayed and opens pop-up window for relevant help page.	Test Successful.

Delete Company				
Test Case	Test	Expected Result	Actual Result	Success/ Failure? (Comment for changes)
Delete action routes to delete company page.	Select the Delete action for a company record.	System routes to delete company page.	System routed to delete company page.	Test Successful.
Page displays delete message.	Attempt to edit caption.	Delete message caption displayed and non- editable.	Delete message caption displayed and non-editable.	Test Successful.
OK Button displayed and routes back to view all companies page where deleted entry non-existent.	Click on OK button.	Button displayed and routes back to view all companies page. Deleted Entry not shown.	Button displayed and routes back to view all companies page. Deleted Entry not shown.	Test Successful.
Cancel Button displayed and routes back to view all companies page. Does not delete entry.	Click on Cancel button.	Button displayed and routes back to view all companies page. No delete.	Button displayed and routes back to view all companies page. No delete.	Test Successful.
Help icon displayed correctly and routes to correct help page.	Click on Help link on page.	Link displayed and opens pop-up window for relevant help page.	Link displayed and opens pop-up window for relevant help page.	Test Successful.

User Object

View All Users				
Test Case	Test	Expected Result	Actual Result	Success? (Comment for changes)
Sidebar routes to view all users page.	Select Users from sidebar menu.	List of the Username, Full Name, Reference fields for all users in table, alphabetical order.	List of the Username, Full Name, Reference fields for all users in table, alphabetical order.	Test Successful.
Links displayed correctly.	Visual verification of links existence.	List of the Username, Full Name, Reference fields for all users in table, alphabetical order.	List of the Username, Full Name, Reference fields for all users in table, alphabetical order.	Test Successful.
Help icon displayed correctly and routes to correct help page.	Click on Help link on page.	Link displayed and opens pop-up window for relevant help page.	Link displayed and opens pop-up window for relevant help page.	Test Successful.

View User				
Test Case	Test	Expected Result	Actual Result	Success/ Failure? (Comment for changes)
View action routes to view user page.	Select the View action for a user record.	System routes to view user page.	System routed to view user page.	Test Successful.
Page displays Username caption.	Attempt to edit caption.	Username caption displayed, non- editable.	Username caption displayed, non-editable.	Test Successful.
Page displays Full Name caption.	Attempt to edit caption.	Full Name caption displayed, non- editable.	Full Name caption displayed, non-editable.	Test Successful.
Page displays Reference caption.	Attempt to edit caption.	Reference caption displayed, non- editable.	Reference caption displayed, non- editable.	Test Successful.
Page displays Extension caption.	Attempt to edit caption.	Extension caption displayed, non- editable.	Extension caption displayed, non- editable.	Test Successful.
Page displays Access Rights caption.	Attempt to edit caption.	Access Rights caption displayed, non- editable.	Access Rights caption displayed, non- editable.	Test Successful.
Page displays Username field.	Attempt to edit field.	Username field displayed, non- editable.	Username field displayed, non-editable.	Test Successful.
Page displays Full Name field.	Attempt to edit field.	Full Name field displayed, non- editable.	Full Name field displayed, non-editable.	Test Successful.

Page displays Reference field.	Attempt to edit field.	Reference field displayed, non- editable.	Reference field displayed, non- editable.	Test Successful.
Page displays Extension field.	Attempt to edit field.	Extension field displayed, non- editable.	Extension field displayed, non- editable.	Test Successful.
Page displays Access Rights field.	Attempt to edit field.	Access Rights field displayed, non- editable.	Access Rights field displayed, non- editable.	Test Successful.
OK Button displayed and routes back to view all users page.	Click on OK button.	Button displayed and routes back to view all users page.	Button displayed and routes back to view all users page.	Test Successful.
Help icon displayed correctly and routes to correct help page.	Click on Help link on page.	Link displayed and opens pop-up window for relevant help page.	Link displayed and opens pop-up window for relevant help page.	Test Successful.

Create User				
Test Case	Test	Expected Result	Actual Result	Success/ Failure? (Comment for changes)
Create New User action routes to view company page.	Select the Create New User action.	System routes to create user page.	System routed to create user page.	Test Successful.
Page displays Username caption.	Attempt to edit caption.	Username caption displayed, non- editable.	Username caption displayed, non-editable.	Test Successful.
Page displays Full Name caption.	Attempt to edit caption.	Full Name caption displayed, non- editable.	Full Name caption displayed, non-editable.	Test Successful.
Page displays Reference caption.	Attempt to edit caption.	Reference caption displayed, non- editable.	Reference caption displayed, non- editable.	Test Successful.
Page displays Extension caption.	Attempt to edit caption.	Extension caption displayed, non- editable.	Extension caption displayed, non- editable.	Test Successful.
Page displays Password caption.	Attempt to edit caption.	Password caption displayed, non- editable.	Password caption displayed, non- editable.	Test Successful.
Page displays Confirm Password caption.	Attempt to edit caption.	Confirm Password caption displayed, non- editable.	Confirm Password caption displayed, non- editable.	Test Successful.
Page displays Access Rights caption.	Attempt to edit caption.	Access Rights caption displayed, non- editable.	Access Rights caption displayed, non- editable.	Test Successful.
Page displays Username text field.	Attempt to edit field.	Field accepts text up to 15 characters.	Field accepts text up to 15 characters.	Test Successful.
Page displays Full Name text field.	Attempt to edit field.	Field accepts text up to 45 characters.	Field accepts text up to 45 characters.	Test Successful.
Page displays Reference text field.	Attempt to edit field.	Field accepts text up to 3 characters.	Field accepts text up to 3 characters.	Test Successful.
Page displays Extension text field.	Attempt to edit field.	Field accepts text up to 6 characters.	Field accepts text up to 6 characters.	Test Successful.
Page displays Password text field.	Attempt to edit field.	Field accepts text up to 20 characters.	Field accepts text up to 20 characters.	Test Successful.
Page displays Confirm Password text field.	Attempt to edit field.	Field accepts text up to 20 characters.	Field accepts text up to 20 characters.	Test Successful.
Page displays Access Rights radio buttons.	Attempt to edit combo-box.	Radio buttons switch successfully.	Radio buttons switch successfully.	Test Successful.
Page displays Username validation error message.	Attempt to submit null field.	Page displays error message for Username Field on submit attempt.	Page displays error message for Username Field on submit attempt.	Test Successful.
Page displays Full Name validation error message.	Attempt to submit null field.	Page displays error message for Full Name Field on submit attempt.	Page displays error message for Full Name Field on submit attempt.	Test Successful.
Page displays Reference validation error message.	Attempt to submit null field.	Page displays error message for Reference Field on submit attempt.	Page displays error message for Reference Field on submit attempt.	Test Successful.
Page displays Password validation error message.	Attempt to submit null field.	Page displays error message for Password Field on submit attempt.	Page displays error message for Password Field on submit attempt.	Test Successful.

Page displays Confirm Password validation error message.	Attempt to submit null field.	Page displays error message for Confirm Password Field on submit attempt.	Page displays error message for Confirm Password Field on submit attempt.	Test Successful.
Page displays Password Match validation error message.	Attempt to submit two different passwords.	Page displays error message if both passwords entered do not match.	Page displays error message if both passwords entered do not match.	Test Successful.
OK Button displayed and routes back to view all users page. Creates new entry.	Click on OK button.	Button displayed and routes back to view all users page. New entry created.	Button displayed and routes back to view all users page. New entry created.	Test Successful.
Cancel Button displayed and routes back to view all users page. Does not create new entry.	Click on Cancel button.	Button displayed and routes back to view all users page. No new entry created.	Button displayed and routes back to view all users page. No new entry created.	Test Successful. INITIAL FAILURE standard radio set to default on page load resolved issue.
Help icon displayed correctly and routes to correct help page.	Click on Help link on page.	Link displayed and opens pop-up window for relevant help page.	Link displayed and opens pop-up window for relevant help page.	Test Successful.

Update User				
Test Case	Test	Expected Result	Actual Result	Success/ Failure? (Comment for changes)
Update User action routes to view company page.	Select the Update User action for a record.	System routes to update user page.	System routed to update user page.	Test Successful.
Page displays Username caption.	Attempt to edit caption.	Username caption displayed, non- editable.	Username caption displayed, non-editable.	Test Successful.
Page displays Full Name caption.	Attempt to edit caption.	Full Name caption displayed, non- editable.	Full Name caption displayed, non-editable.	Test Successful.
Page displays Reference caption.	Attempt to edit caption.	Reference caption displayed, non- editable.	Reference caption displayed, non- editable.	Test Successful.
Page displays Extension caption.	Attempt to edit caption.	Extension caption displayed, non- editable.	Extension caption displayed, non- editable.	Test Successful.
Page displays Password caption.	Attempt to edit caption.	Password caption displayed, non- editable.	Password caption displayed, non- editable.	Test Successful.
Page displays Confirm Password caption.	Attempt to edit caption.	Confirm Password caption displayed, non- editable.	Confirm Password caption displayed, non- editable.	Test Successful.
Page displays Access Rights caption.	Attempt to edit caption.	Access Rights caption displayed, non- editable.	Access Rights caption displayed, non- editable.	Test Successful.
Page displays Username text field contents from database.	Update text field.	Displays field contents from database. Username text field accepts text up to 15 characters.	Displays field contents from database. Username text field accepts text up to 15 characters.	Test Successful.
Page displays Full Name text field contents from database.	Update text field.	Displays field contents from database. Full Name text field accepts text up to 45 characters.	Displays field contents from database. Full Name text field accepts text up to 45 characters.	Test Successful.
Page displays Reference text field contents from database.	Update text field.	Displays field contents from database. Reference text field accepts text up to 3 characters.	Displays field contents from database. Reference text field accepts text up to 3 characters.	Test Successful.
Page displays Extension text field contents from database.	Update text field.	Displays field contents from database. Extension text field accepts text up to 6 characters.	Displays field contents from database. Extension text field accepts text up to 6 characters.	Test Successful.
Page displays Password text field contents from database.	Update text field.	Displays field contents from database. Password text field accepts text up to 20 characters.	Displays field contents from database. Password text field accepts text up to 20 characters.	Test Successful.
Page displays Confirm	Update text field.	Displays field contents from	Displays field contents from	Test Successful.

Password text field contents from database.		database. Confirm Password text field accepts text up to 20 characters.	database. Confirm Password text field accepts text up to 20 characters.	
Page displays Access Rights radio buttons.	Update radio buttons.	Displays value from database. Allows update of radio button.	Displays value from database. Allows update of radio button.	Test Successful.
Page displays Username validation error message.	Attempt to submit null field.	Page displays error message for Username Field on submit attempt.	Page displays error message for Username Field on submit attempt.	Test Successful.
Page displays Full Name validation error message.	Attempt to submit null field.	Page displays error message for Full Name Field on submit attempt.	Page displays error message for Full Name Field on submit attempt.	Test Successful.
Page displays Reference validation error message.	Attempt to submit null field.	Page displays error message for Reference Field on submit attempt.	Page displays error message for Reference Field on submit attempt.	Test Successful.
Page displays Password validation error message.	Attempt to submit null field.	Page displays error message for Password Field on submit attempt.	Page displays error message for Password Field on submit attempt.	Test Successful.
Page displays Confirm Password validation error message.	Attempt to submit null field.	Page displays error message for Confirm Password Field on submit attempt.	Page displays error message for Confirm Password Field on submit attempt.	Test Successful.
Page displays Password Match validation error message.	Attempt to submit two different passwords.	Page displays error message if both passwords entered do not match.	Page displays error message if both passwords entered do not match.	Test Successful.
OK Button displayed and routes back to view all users page. Creates new entry.	Click on OK button.	Button displayed and routes back to view all users page. New entry created.	Button displayed and routes back to view all users page. New entry created.	Test Successful.
Cancel Button displayed and routes back to view all users page. Does not create new entry.	Click on Cancel button.	Button displayed and routes back to view all users page. No new entry created.	Button displayed and routes back to view all users page. No new entry created.	Test Successful. INITIAL FAILURE standard radio set to default on page load resolved issue.
Help icon displayed correctly and routes to correct help page.	Click on Help link on page.	Link displayed and opens pop-up window for relevant help page.	Link displayed and opens pop-up window for relevant help page.	Test Successful.

Delete User				
Test Case	Test	Expected Result	Actual Result	Success/ Failure? (Comment for changes)
Delete action routes to delete user page.	Select the Delete action for a user record.	System routes to delete user page.	System routed to delete user page.	Test Successful.
Page displays delete message.	Attempt to edit caption.	Delete message caption displayed and non- editable.	Delete message caption displayed and non-editable.	Test Successful.
OK Button displayed and routes back to view all users page where deleted entry non-existent.	Click on OK button.	Button displayed and routes back to view all users page. Deleted Entry not shown.	Button displayed and routes back to view all users page. Deleted Entry not shown.	Test Successful.
Cancel Button displayed and routes back to view all users page. Does not delete entry.	Click on Cancel button.	Button displayed and routes back to view all users page. No delete.	Button displayed and routes back to view all users page. No delete.	Test Successful.
Help icon displayed correctly and routes to correct help page.	Click on Help link on page.	Link displayed and opens pop-up window for relevant help page.	Link displayed and opens pop-up window for relevant help page.	Test Successful.

Access Object

System Login				
Test Case	Test	Expected Result	Actual Result	Success? (Comment for changes)
Sidebar routes to login page.	Select Login from sidebar menu.	Login Page with Username and password fields, OK and Cancel button.	Login Page with Username and password fields, OK and Cancel button.	Test Successful.
Buttons displayed correctly.	Visual verification of buttons.	OK and Cancel button shown under text field.	OK and Cancel button shown under text field.	Test Successful.
Page displays Username caption.	Attempt to edit caption.	Username caption displayed, non- editable.	Username caption displayed, non-editable.	Test Successful.
Page displays Password caption.	Attempt to edit caption.	Password caption displayed, non- editable.	Password caption displayed, non- editable.	Test Successful.
Page displays Username text field.	Attempt to edit field.	Field accepts text up to 15 characters.	Field accepts text up to 15 characters.	Test Successful.
Page displays Password text field.	Attempt to edit field.	Field accepts text up to 20 characters.	Field accepts text up to 20 characters.	Test Successful.
Page displays Username validation error message.	Attempt to submit null field.	Page displays error message for Username Field on submit attempt.	Page displays error message for Username Field on submit attempt.	Test Successful.
Page displays Password validation error message.	Attempt to submit null field.	Page displays error message for Password Field on submit attempt.	Page displays error message for Password Field on submit attempt.	Test Successful.
Page displays Invalid login validation error message.	Attempt to submit invalid login details.	Page displays Invalid login validation error message.	Page displays Invalid login validation error message.	Test Successful.
Page displays Invalid login validation error message.	Attempt to submit invalid user who has been deleted.	Page displays Invalid login validation error message.	Page displays Invalid login validation error message.	Test Successful.
Login Button displayed and routes back to view orders page where deleted entry non-existent.	Click on Login button.	Button displayed and routes back to view orders page. Delete Entry not shown.	Button displayed and routes back to view orders page. Delete Entry not shown.	Test Successful. INITIAL FAILURE incorrect router action resolved
Cancel Button displayed and does not login user.	Click on Cancel button.	Cancel Button displayed and does not login user.	Cancel Button displayed and does not login user.	Test Successful.
Header displays Full Name and access rights of user logged in.	Visual check of header.	Header displays Full Name and access rights of user logged in.	Header displays Full Name and access rights of user logged in.	Test Successful.
Sidebar Login caption changes to Logout.	Visual check of sidebar.	Sidebar Login caption changes to Logout.	Sidebar Login caption changes to Logout.	Test Successful.
Help icon displayed correctly and routes to correct help page.	Click on Help link on page.	Link displayed and opens pop-up window for relevant help page.	Link displayed and opens pop-up window for relevant help page.	Test Successful.

System Logout				
Test Case	Test	Expected Result	Actual Result	Success? (Comment for changes)
Sidebar routes to login page.	Select Logout from sidebar menu.	Login Page with Username and password fields, OK and Cancel button.	Login Page with Username and password fields, OK and Cancel button.	Test Successful.
Links displayed correctly.	Visual verification of buttons.	Logout link shown in sidebar.	Logout link shown in sidebar.	Test Successful.
Header login info disappears.	Visual check of header.	Header does not display Full Name and access rights of user logged in.	Header does not display Full Name and access rights of user logged in.	Test Successful. INITIAL FAILURE cleaned up session correctly.
Sidebar Logout caption changes to Login.	Visual check of sidebar.	Sidebar Logout caption changes to Login.	Sidebar Logout caption changes to Login.	Test Successful.

Determining Access Rights				
Test Case	Test	Expected Result	Actual Result	Success? (Comment for changes)
Attempt to View User without access permission set.	Select View user	Feature Access Denied message.	Feature Access Denied message.	Test Successful. INITIAL FAILURE process action corrected.
Attempt to Create User without access permission set.	Select Create New user	Feature Access Denied message.	Feature Access Denied message.	Test Successful.
Attempt to Update User without access permission set.	Select Update for a user record.	Feature Access Denied message.	Feature Access Denied message.	Test Successful.
Attempt to Delete User without access permission.	Select Delete for a user record.	Feature Access Denied message.	Feature Access Denied message.	Test Successful.
Attempt to View Company without access permission set.	Select View Company	Feature Access Denied message.	Feature Access Denied message.	Test Successful. INITIAL FAILURE process action corrected.
Attempt to Create Company without access permission set.	Select Create New Company	Feature Access Denied message.	Feature Access Denied message.	Test Successful.
Attempt to Update Company without access permission set.	Select Update for a Company record.	Feature Access Denied message.	Feature Access Denied message.	Test Successful.
Attempt to Delete Company without access permission set.	Select Delete for a Company record.	Feature Access Denied message.	Feature Access Denied message.	Test Successful.
Attempt to View Order without access permission set.	Select View Order	Feature Access Denied message.	Feature Access Denied message.	Test Successful. INITIAL FAILURE process action corrected.
Attempt to Create Order without access permission set.	Select Create New Order	Feature Access Denied message.	Feature Access Denied message.	Test Successful.
Attempt to Update Order without access permission set.	Select Update for an Order record.	Feature Access Denied message.	Feature Access Denied message.	Test Successful.
Attempt to Delete Order without access permission set.	Select Delete for an Order record.	Feature Access Denied message.	Feature Access Denied message.	Test Successful.
Attempt to Create Order-line without access permission set.	Select Create New Order-line	Feature Access Denied message.	Feature Access Denied message.	Test Successful.
Attempt to Delete Order-line without access permission set.	Select Delete for an Order-line record.	Feature Access Denied message.	Feature Access Denied message.	Test Successful.
Attempt to Finish Order without access permission set.	Select Finish Order for an order record.	Feature Access Denied message.	Feature Access Denied message.	Test Successful.
Attempt to Check-In Order without access permission set.	Select Check-In for an Order record.	Feature Access Denied message.	Update page is shown but without ability to change access rights.	Test Successful.
Access Rights radio buttons not shown for non-admin user.	Visual Check of Update user page.	Update page is shown but without ability to change access rights.	Update page is shown but without ability to change access rights.	Test Successful.
Access Link in sidebar not shown for non-admin user.	Visual Check of sidebar.	Access Link in sidebar not shown for non-admin user.	Access Link in sidebar not shown for non-admin user.	Test Successful.
Attempt to View User with access permission set.	Select View user	Routes to correct page.	Routes to correct page.	Test Successful.

Attempt to Create User with access permission set.	Select Create New user	Routes to correct page.	Routes to correct page.	Test Successful.
Attempt to Update User with access permission set.	Select Update for a user record.	Routes to correct page.	Routes to correct page.	Test Successful.
Attempt to Delete User with access permission.	Select Delete for a user record.	Routes to correct page.	Routes to correct page.	Test Successful.
Attempt to View Company with access permission set.	Select View Company	Routes to correct page.	Routes to correct page.	Test Successful.
Attempt to Create Company with access permission set.	Select Create New Company	Routes to correct page.	Routes to correct page.	Test Successful.
Attempt to Update Company with access permission set.	Select Update for a Company record.	Routes to correct page.	Routes to correct page.	Test Successful.
Attempt to Delete Company with access permission set.	Select Delete for a Company record.	Routes to correct page.	Routes to correct page.	Test Successful.
Attempt to View Order with access permission set.	Select View Order	Routes to correct page.	Routes to correct page.	Test Successful. INITIAL FAILURE process action corrected.
Attempt to Create Order with access permission set.	Select Create New Order	Routes to correct page.	Routes to correct page.	Test Successful.
Attempt to Update Order with access permission set.	Select Update for an Order record.	Routes to correct page.	Routes to correct page.	Test Successful.
Attempt to Delete Order with access permission set.	Select Delete for an Order record.	Routes to correct page.	Routes to correct page.	Test Successful.
Attempt to Create Order-line with access permission set.	Select Create New Order-line	Routes to correct page.	Routes to correct page.	Test Successful.
Attempt to Delete Order-line with access permission set.	Select Delete for an Order-line record.	Deletes order-line and recalculates total.	Deletes order-line and recalculates total.	Test Successful.
Attempt to Finish Order with access permission set.	Select Finish Order for an order record.	Routes to correct page.	Routes to correct page.	Test Successful.
Attempt to Check-In Order with access permission set.	Select Check-In for an Order record.	Feature Access Denied message.	Update page is shown but without ability to change access rights.	Test Successful.
Access Rights radio buttons not shown for non-admin user.	Visual Check of Update user page.	Update page is shown but without ability to change access rights.	Update page is shown but without ability to change access rights.	Test Successful.
Access Link in sidebar not shown for non-admin user.	Visual Check of sidebar.	Access Link in sidebar not shown for non-admin user.	Access Link in sidebar not shown for non-admin user.	Test Successful.
OK Button displayed and routes back to view all orders page. Updates access.	Click on OK button.	Button displayed and routes back to view all orders page. Updates access.	Button displayed and routes back to view all orders page. Updates access.	Test Successful.
Cancel Button displayed and routes back to view all orders page. Does not update access.	Click on Cancel button.	Button displayed and routes back to view all orders page. No update.	Button displayed and routes back to view all orders page. No update.	Test Successful.

Order Object

View Orders Summary				
Test Case	Test	Expected Result	Actual Result	Success? (Comment for

				changes)
Sidebar routes to order summary page.	Select Orders from sidebar menu.	List of the order no, company, date required for all confirmed and unconfirmed orders.	List of the order no, company, date required for all confirmed and unconfirmed orders.	Test Successful. INITIAL FAILURE to set confirmed resolved.
Links displayed correctly for unconfirmed orders.	Visual verification of links existence.	Each unconfirmed order has Update and Delete action links.	Each unconfirmed order has Update and Delete action links.	Test Successful.
Links displayed correctly for confirmed orders.	Visual verification of links existence.	Each confirmed order has View, Check-in and Finish action links.	Each confirmed order has View, Check-in and Finish action links.	Test Successful.
Buttons all exist and displayed correctly.	Visual verification of button existence.	Create order button next to companies combo-box	Create order button next to companies combo-box	Test Successful.
Companies' combo-box populated with companies from DB.	Select arrow to reveal companies.	Companies in alphabetical order in combo-box.	Companies in alphabetical order in combo-box.	Test Successful.

Create Order				
Test Case	Test	Expected Result	Actual Result	Success/ Failure? (Comment for changes)
Create New order action routes to create order page.	Select the Create New Order action.	System routes to create order page.	System routed to create order page.	Test Successful.
Page displays Code caption.	Attempt to edit caption.	Code caption displayed, non-editable.	Code caption displayed, non-editable.	Test Successful.
Page displays Address delivery method caption.	Attempt to edit caption.	Delivery Method caption displayed, non- editable.	Delivery Method caption displayed, non- editable.	Test Successful.
Page displays date required caption.	Attempt to edit caption.	Date required caption displayed, non- editable.	Date required caption displayed, non- editable.	Test Successful.
Page displays details caption.	Attempt to edit caption.	Details caption displayed, non- editable.	Details caption displayed, non- editable.	Test Successful.
Page displays Confirm caption. (access dependant)	Attempt to edit caption.	Confirm caption displayed for admin users, non- editable.	Confirm caption displayed for admin users, non- editable.	Test Successful.
Page displays Order line captions (9).	Attempt to edit captions.	Captions displayed, non-editable. Action also shown,	Captions displayed, non-editable. Action also shown,	Test Successful.
Page displays Code field.	Attempt to edit field.	Code field displayed, non-editable. Auto-generate code	Code field displayed, non-editable. Auto-generate code	Test Successful.
Page displays Delivery method combo-box.	Enter text into field.	Combo box with 2 items. Standard delivery default on page load.	Combo box with 2 items. Standard delivery default on page load.	Test Successful.
Page displays Date required text field.	Enter text into field.	15 chars max of the format DD-MM-YYYY.	15 chars max of the format DD-MM-YYYY.	Test Successful. INITIAL FAILURE DB date format resolved.
Confirm order allows check.	Check Box.	Allows checking of box.	Allows checking of box.	Test Successful.
Page displays Date required validation error message.	Attempt to submit null field.	Page displays error message for Date required Field on submit attempt.	Page displays error message for Date required Field on submit attempt.	Test Successful.
Page displays date required validation error message.	Attempt to submit invalid date.	Page displays error message for Date required Field on submit attempt.	Page displays error message for Date required Field on submit attempt.	Test Successful.
OK Button displayed and routes back to view all orders page. Creates new entry.	Click on OK button.	Button displayed and routes back to view all orders page. New entry created. Confirmed if applicable.	Button displayed and routes back to view all orders page. New entry created. Confirmed if applicable.	Test Successful.
Cancel Button displayed and routes back to view all orders page. Does not create new entry.	Click on Cancel button.	Button displayed and routes back to view all orders page. No new entry created.	Button displayed and routes back to view all orders page. No new entry created.	Test Successful.
Help icon displayed	Click on Help link on	Link displayed and opens	Link displayed and opens	Test Successful.

correctly and routes to correct help page.	page.	pop-up window for relevant help page.	pop-up window for relevant help page.	

Update Order				
Test Case	Test	Expected Result	Actual Result	Success/ Failure? (Comment for changes)
Update order action routes to create order page.	Select the update Order action.	System routes to update order page.	System routed to update order page.	Test Successful.
Page displays Code caption.	Attempt to edit caption.	Code caption displayed, non-editable.	Code caption displayed, non-editable.	Test Successful.
Page displays Address delivery method caption.	Attempt to edit caption.	Delivery Method caption displayed, non- editable.	Delivery Method caption displayed, non- editable.	Test Successful.
Page displays date required caption.	Attempt to edit caption.	Date required caption displayed, non- editable.	Date required caption displayed, non- editable.	Test Successful.
Page displays details caption.	Attempt to edit caption.	Details caption displayed, non- editable.	Details caption displayed, non- editable.	Test Successful.
Page displays Confirm caption. (access dependant)	Attempt to edit caption.	Confirm caption displayed for admin users, non- editable.	Confirm caption displayed for admin users, non- editable.	Test Successful.
Page displays Order line captions (9).	Attempt to edit captions.	Captions displayed, non-editable. Action also shown,	Captions displayed, non-editable. Action also shown,	Test Successful.
Page displays Code field.	Attempt to edit field.	Code field displayed, non-editable. From DB.	Code field displayed, non-editable. From DB.	Test Successful.
Page displays Delivery method combo-box.	Enter text into field.	Combo box with 2 items. Returns value from record.	Combo box with 2 items. Returns value from record.	Test Successful.
Page displays Date required text field.	Enter text into field.	15 chars max of the format DD-MM-YYYY. Returns value from record.	15 chars max of the format DD-MM-YYYY. Returns value from record.	Test Successful. INITIAL FAILURE DB date format resolved.
Confirm order allows check.	Check Box.	Allows checking of box.	Allows checking of box.	Test Successful.
Page displays Date required validation error message.	Attempt to submit null field.	Page displays error message for Date required Field on submit attempt.	Page displays error message for Date required Field on submit attempt.	Test Successful.
Page displays date required validation error message.	Attempt to submit invalid date.	Page displays error message for Date required Field on submit attempt.	Page displays error message for Date required Field on submit attempt.	Test Successful.
OK Button displayed and routes back to view all orders page. Creates new entry.	Click on OK button.	Button displayed and routes back to view all orders page. New entry updated. Confirmed if applicable.	Button displayed and routes back to view all orders page. New entry updated. Confirmed if applicable.	Test Successful.
Cancel Button displayed and routes back to view all orders page. Does not create new entry.	Click on Cancel button.	Button displayed and routes back to view all orders page. No entry updated.	Button displayed and routes back to view all orders page. No entry updated.	Test Successful.
Help icon displayed correctly and routes to correct help page.	Click on Help link on page.	Link displayed and opens pop-up window for relevant help page.	Link displayed and opens pop-up window for relevant help page.	Test Successful.

Delete Order				
Test Case	Test	Expected Result	Actual Result	Success/ Failure? (Comment for changes)
Delete action routes to delete order page.	Select the Delete action for an order record.	System routes to delete order page.	System routed to delete order page.	Test Successful.
Page displays delete	Attempt to edit	Delete message caption	Delete message caption	Test Successful.

message.	caption.	displayed and non- editable.	displayed and non-editable.	
OK Button displayed and routes back to view all orders page where deleted entry non-existent.	Click on OK button.	Button displayed and routes back to view all orders page. Deleted Entry not shown.	Button displayed and routes back to view all orders page. Deleted Entry not shown.	Test Successful.
Cancel Button displayed and routes back to view all orders page. Does not delete entry.	Click on Cancel button.	Button displayed and routes back to view all orders page. No delete.	Button displayed and routes back to view all orders page. No delete.	Test Successful.
Help icon displayed correctly and routes to correct help page.	Click on Help link on page.	Link displayed and opens pop-up window for relevant help page.	Link displayed and opens pop-up window for relevant help page.	Test Successful.

Finish Order				
Test Case	Test	Expected Result	Actual Result	Success/ Failure? (Comment for changes)
Delete action routes to finish order page.	Select the finish action for an order record.	System routes to finish order page.	System routed to finish order page.	Test Successful.
Page displays delete message.	Attempt to edit caption.	Delete message caption displayed and non- editable.	Delete message caption displayed and non-editable.	Test Successful.
OK Button displayed and routes back to view all orders page where deleted entry non-existent.	Click on OK button.	Button displayed and routes back to view all orders page. Deleted Entry not shown.	Button displayed and routes back to view all orders page. Deleted Entry not shown.	Test Successful.
Cancel Button displayed and routes back to view all orders page. Does not delete entry.	Click on Cancel button.	Button displayed and routes back to view all orders page. No delete.	Button displayed and routes back to view all orders page. No delete.	Test Successful. INITIAL FAILURE repaired object and JSP page– all OK
Help icon displayed correctly and routes to correct help page.	Click on Help link on page.	Link displayed and opens pop-up window for relevant help page.	Link displayed and opens pop-up window for relevant help page.	Test Successful.

View Confirmed Order				
Test Case	Test	Expected Result	Actual Result	Success/ Failure? (Comment for changes)
View action routes to delete order page.	Select the View action for an order record.	System routes to PDF access object.	System routes to PDF access object.	Test Successful.
Page displays order in PDF window.	Attempt to edit caption.	Page displays order in PDF window.	Page displays order in PDF window.	Test Successful. INITIAL FAILURE JavaScript repaired
PDF displays correct order, customer, order-lines and additional info.	Examine PDF contents.	PDF displays correct order, customer, order-lines and additional info	PDF displays correct order, customer, order-lines and additional info	Test Successful. INITIAL FAILURE populating order-lines – all now OK.

Create Order Line				
Test Case	Test	Expected Result	Actual Result	Success/ Failure? (Comment for changes)
Create Order Line action routes to create order line page.	Select the Order Line action.	System routes to Order Line page.	System routes to Order Line page.	Test Successful.
Page displays Supplier caption.	Attempt to edit caption.	Supplier caption displayed, non- editable.	Supplier caption displayed, non- editable.	Test Successful.
Page displays Quantity caption.	Attempt to edit caption.	Quantity caption displayed, non- editable.	Quantity caption displayed, non- editable.	Test Successful.

Page displays Code caption.	Attempt to edit caption.	Code caption displayed, non-editable.	Code caption displayed, non-editable.	Test Successful.
Page displays Size caption.	Attempt to edit caption.	Size caption displayed, non-editable.	Size caption displayed, non-editable.	Test Successful.
Page displays Colour caption.	Attempt to edit caption.	Colour caption displayed, non- editable.	Colour caption displayed, non-editable.	Test Successful.
Page displays Description caption.	Attempt to edit caption.	Description caption displayed, non- editable.	Description caption displayed, non- editable.	Test Successful.
Page displays Unit Price caption.	Attempt to edit caption.	Unit Price caption displayed, non- editable.	Unit Price caption displayed, non- editable.	Test Successful.
Page displays Supplier combo-box.	Enter text into field.	Contains list of Suppliers to choose	Contains list of Suppliers to choose	Test Successful. INITIAL FAILURE modified object to return suppliers.
Page displays Quantity text field.	Enter text into field.	Quantity text field accepts text up to 11 characters.	Quantity text field accepts text up to 11 characters.	Test Successful.
Page displays Code text field.	Enter text into field.	Code text field accepts text up to 30 characters.	Code text field accepts text up to 30 characters.	Test Successful.
Page displays Size text field.	Enter text into field.	Size text field accepts text up to 30 characters.	Size text field accepts text up to 30 characters.	Test Successful.
Page displays Colour text field.	Enter text into field.	Colour text field accepts text up to 30 characters.	Colour text field accepts text up to 30 characters.	Test Successful.
Page displays Description text field.	Enter text into field.	Description text field accepts text up to 30 characters.	Description text field accepts text up to 30 characters.	Test Successful.
Page displays Unit Price text field.	Enter numbers into field.	Postcode text field accepts text up to 11 digits.	Postcode text field accepts text up to 11 digits.	Test Successful.
Page displays Quantity validation error message.	Attempt to submit null field.	Page displays error message for Quantity Field on submit attempt.	Page displays error message for Quantity Field on submit attempt.	Test Successful.
Page displays Code validation error message.	Attempt to submit null field.	Page displays error message for Code Field on submit attempt.	Page displays error message for Code Field on submit attempt.	Test Successful.
Page displays Size validation error message.	Attempt to submit null field.	Page displays error message for Size Field on submit attempt.	Page displays error message for Size Field on submit attempt.	Test Successful.
Page displays Colour validation error message.	Attempt to submit null field.	Page displays error message for Colour Field on submit attempt.	Page displays error message for Colour Field on submit attempt.	Test Successful.
Page displays Description validation error message.	Attempt to submit null field.	Page displays error message for Description Field on submit attempt.	Page displays error message for Description Field on submit attempt.	Test Successful.
Page displays Unit Price validation error message.	Attempt to submit null field.	Page displays error message for Unit Price Field on submit attempt.	Page displays error message for Unit Price Field on submit attempt.	Test Successful.
OK Button displayed and routes back to update order page. Creates new entry.	Click on OK button.	Button displayed and routes back to update order page. New entry created. Total adjusted.	Button displayed and routes back to update order page. New entry created. Total adjusted.	Test Successful. INITIAL FAILURE to recalculate now all OK.
Cancel Button displayed and routes back to view all companies page. Does not create new entry.	Click on Cancel button.	Button displayed and routes back to update order page. No new entry created. Total stays the same.	Button displayed and routes back to update order page. No new entry created. Total stays the same.	Test Successful.
Help icon displayed correctly and routes to correct help page.	Click on Help link on page.	Link displayed and opens pop-up window for relevant help page.	Link displayed and opens pop-up window for relevant help page.	Test Successful.

Delete Order Line				
Test Case	Test	Expected Result	Actual Result	Success/ Failure? (Comment for changes)
Page displays delete action for each order	Attempt to edit caption.	Page displays delete action for each order line.	Page displays delete action for each order line.	Test Successful.

line.				
Deletes order line and recalculates total.	Click link to delete order line.	Deletes order line and recalculates total.	Deletes order line and recalculates total.	Test Successful. INITIAL FAILURE when routing to other page. All OK.

Check-In Items				
Test Case	Test	Expected Result	Actual Result	Success/ Failure? (Comment for changes)
Update Check-In action routes to update check-in page.	Select the Check-In action for a confirmed order.	System routes to update check-in page.	System routes to update check-in page.	Test Successful.
Page displays Code caption.	Attempt to edit caption.	Code caption displayed, non-editable.	Code caption displayed, non-editable.	Test Successful.
Page displays name caption.	Attempt to edit caption.	Name caption displayed, non- editable.	Name caption displayed, non-editable.	Test Successful.
Page displays Address caption.	Attempt to edit caption.	Address caption displayed, non- editable.	Address caption displayed, non- editable.	Test Successful.
Page displays Telephone caption.	Attempt to edit caption.	Telephone caption displayed, non- editable.	Telephone caption displayed, non- editable.	Test Successful.
Page displays Code field.	Attempt to edit field.	Code field displayed, non-editable.	Code field displayed, non-editable.	Test Successful.
Page displays name field.	Attempt to edit field.	Name field displayed, non-editable.	Name field displayed, non-editable.	Test Successful.
Page displays Address field.	Attempt to edit field.	Address field displayed, non-editable.	Address field displayed, non-editable.	Test Successful.
Page displays Telephone field.	Attempt to edit field.	Telephone field displayed, non- editable.	Telephone field displayed, non- editable.	Test Successful.
Page displays Quantity validation error message.	Attempt to submit null field.	Page displays error message for Quantity Field on submit attempt.	Page displays error message for Quantity Field on submit attempt.	Test Successful.
Page displays Comment validation error message.	Attempt to submit null field.	Page displays error message for Comment Field on submit attempt.	Page displays error message for Comment Field on submit attempt.	Test Successful.
OK Button displayed and routes back to order page. Updates entry.	Click on OK button.	Button displayed and routes back to orders page. Total adjusted. Image updates.	Button displayed and routes back to orders page. Total adjusted. Image updates.	Test Successful. INITIAL FAILURE to display image all OK.
Cancel Button displayed and routes back to orders page. Does not update entry.	Click on Cancel button.	Button displayed and routes back to update order page. No update. Quantity stays the same.	Button displayed and routes back to update order page. No update. Quantity stays the same.	Test Successful.
Help icon displayed correctly and routes to correct help page.	Click on Help link on page.	Link displayed and opens pop-up window for relevant help page.	Link displayed and opens pop-up window for relevant help page.	Test Successful.

Reports Object

Perform Search				
Test Case	Test	Expected Result	Actual Result	Success/ Failure? (Comment for changes)
Perform search routes to update check-in page.	Select the Perform search link.	System routes to Perform search page.	System routes to Perform search page.	Test Successful.
Page displays Customer	Attempt to edit	Customer caption displayed,	Customer caption displayed,	Test Successful.

caption.	caption.	non- editable.	non- editable.	
Page displays Supplier caption.	Attempt to edit caption.	Supplier caption displayed, non- editable.	Supplier caption displayed, non- editable.	Test Successful.
Page displays Sales Person caption.	Attempt to edit caption.	Sales Person caption displayed, non- editable.	Sales Person caption displayed, non- editable.	Test Successful.
Page displays Date From caption.	Attempt to edit caption.	Date From caption displayed, non- editable.	Date From caption displayed, non- editable.	Test Successful.
Page displays Sales Date To caption.	Attempt to edit caption.	Date To caption displayed, non- editable.	Date To caption displayed, non- editable.	Test Successful.
Page displays Customer combo-box.	Attempt to edit combo-box.	Selectable Item from combo-box.	Selectable Item from combo	Test Successful.
Page displays Supplier combo-box.	Attempt to edit combo-box.	Selectable Item from combo	Selectable Item from combo	Test Successful.
Page displays Sales Person combo-box.	Attempt to edit combo-box.	Selectable Item from combo	Selectable Item from combo	Test Successful.
Page displays Date From field.	Attempt to edit field.	Text field accepts text up to 11 characters.	Text field accepts text up to 11 characters.	Test Successful.
Page displays Sales Date To field.	Attempt to edit field.	Text field accepts text up to 11 characters.	Text field accepts text up to 11 characters.	Test Successful.
Page displays Date From validation error message.	Attempt to submit null field.	Page displays error message for Date From Field on submit attempt.	Page displays error message for Date From Field on submit attempt.	Test Successful.
Page displays Date to validation error message.	Attempt to submit null field.	Page displays error message for Date to Field on submit attempt.	Page displays error message for Date to Field on submit attempt.	Test Successful.
Page displays Date From validation error message.	Attempt to submit date not in format of DD-MM-YYYY	Page displays error message for Date From Field on submit attempt.	Page displays error message for Date From Field on submit attempt.	Test Successful.
Page displays Date to validation error message.	Attempt to submit date not in format of DD-MM-YYYY	Page displays error message for Date to Field on submit attempt.	Page displays error message for Date to Field on submit attempt.	Test Successful.
Search Button displayed and performs search for matching criteria.	Click on Search button.	Results displayed with table showing code, date and view action for order.	Results displayed with table showing code, date and view action for order.	Test Successful.
Search Button displayed and performs search for no matching criteria.	Click on Search button.	Search results no match message displayed.	Search results no match message displayed.	Test Successful. INITIAL FAILURE to display message all OK
Help icon displayed correctly and routes to correct help page.	Click on Help link on page.	Link displayed and opens pop-up window for relevant help page.	Link displayed and opens pop-up window for relevant help page.	Test Successful.

Appendix N – Black Box Testing Results

Test Case	Expected Result	Actual Result	Success?
Start System	Welcome page shown with company logo	Welcome page shown with company logo	Test Successful
Login as Administrator	Sidebar shows all navigation options. Header info updates for admin user.	Sidebar shows all navigation options. Header info updates for admin user.	Test Successful
Login as Standard User	Sidebar shows all but Access Rights navigation options. Header info updates for standard user	Sidebar shows all but Access Rights navigation options. Header info updates for standard user	Test Successful
Login as Temporary User.	Sidebar shows all but Access Rights navigation options	Sidebar shows all but Access Rights navigation options	Test Successful
Attempt to navigate after session timer expiry.	Routed to login page	Routed to login page	Test Successful
Attempt to navigate to a page without logging into system	Routed to login page	Routed to login page	Test Successful
User logging out of system.	Routed to login page. Header login info disappears	Routed to login page. Header login info disappears	Test Successful
View all Companies as an administrator	Companies page loads and all links allow access	Companies page loads and all links allow access	Test Successful
View all Companies as a Standard User	Companies page loads but delete link denies access	Companies page loads but delete link denies access	Test Successful
View all Companies as a Temporary User	Companies page loads but only view link permits access	Companies page loads but only view link permits access	Test Successful
View all Users as an administrator	Users page loads and all links allow access	Users page loads and all links allow access	Test Successful
View all Users as a Standard User	Users page loads but delete link denies access	Users page loads but delete link denies access	Test Successful
View all Users as a Temporary User	Users page loads but only view link permits access.	Users page loads but only view link permits access.	Test Successful
View all Orders as an administrator.	Orders page loads and all links allow access	Orders page loads and all links allow access	Test Successful
View all Orders as a Standard User	Orders page loads but delete, finish and create links deny access	Orders page loads but delete, finish and create links deny access	Test Successful
View all Orders as a Temporary User	Orders page loads but only view link permits access	Orders page loads but only view link permits access	Test Successful
View Reports	The reports search facility page appears	The reports search facility page appears	Test Successful
Create a New Company	New company process pursued. New company shown in all companies view	New company process pursued. New company shown in all companies view	Test Successful
Update Company	Update company process	Update company process	Test Successful

	pursued. Update evident for those fields shown in view all companies	pursued. Update evident for those fields shown in view all companies	
Delete Company	Delete company process pursued. Delete evident for those fields shown in view all companies	Delete company process pursued. Delete evident for those fields shown in view all companies	Test Successful
Create a New User	New user process pursued. New company shown in all companies view	New user process pursued. New company shown in all companies view	Test Successful
Update User (admin user)	Update user process pursued. Update evident for those fields shown in view all companies. Access rights buttons shown	Update user process pursued. Update evident for those fields shown in view all companies. Access rights buttons shown	Test Successful
Update User (non admin user)	Update user process pursued. Update evident for those fields shown in view all companies. Access rights buttons not shown	Update user process pursued. Update evident for those fields shown in view all companies. Access rights buttons not shown	Test Successful
Delete User	Delete user process pursued. Delete evident for those fields shown in view all companies	Delete user process pursued. Delete evident for those fields shown in view all companies	Test Successful
Create a New Order	New Order process pursued. Auto-generate order number. New order shown in orders view	New Order process pursued. Auto-generate order number. New order shown in orders view	Test Successful
Update Order (admin user)	Update order process pursued. Update evident for those fields shown in view all orders. Confirm order check-box shown	Update order process pursued. Update evident for those fields shown in view all orders. Confirm order check-box shown	Test Successful
Update Order (non-admin user)	Update order process pursued. Update evident for those fields shown in view all orders. Confirm order check-box not shown	Update order process pursued. Update evident for those fields shown in view all orders. Confirm order check-box not shown	Test Successful
Delete Order	Delete order process pursued. Delete evident for those fields shown in view all order	Delete order process pursued. Delete evident for those fields shown in view all order	Test Successful
Finish Order	Finish order process pursued. Finish evident for those fields shown in view all order	Finish order process pursued. Finish evident for those fields shown in view all order	Test Successful
Check-In Order	Check-In process pursued. Routed back to view all orders screen	Check-In process pursued. Routed back to view all orders screen	Test Successful
View Order	Relevant PDF order shown in popup window	Relevant PDF order shown in popup window	Test Successful
Create Order Line	New order line process pursued. New order line shown in update order view. Order total recalculate	New order line process pursued. New order line shown in update order view. Order total recalculate	Test Successful

Delete Order Line	Delete Order Line process pursued. Order total recalculate. Order line disappears from update order view	Delete Order Line process pursued. Order total recalculate. Order line disappears from update order view	Test Successful
Check-In items over quantified	Order Line in Check-In process shows 'arrow-up' image for appropriate over-quantified order line	Order Line in Check-In process shows 'arrow-up' image for appropriate over-quantified order line	Test Successful
Check-In items under quantified	Order Line in Check-In process shows 'arrow-down' image for appropriate under-quantified order line	Order Line in Check-In process shows 'arrow-down' image for appropriate under-quantified order line	Test Successful
Check-In items zero quantified	Order Line in Check-In process shows 'X' image for appropriate zero-quantified order line	Order Line in Check-In process shows 'X' image for appropriate zero-quantified order line	Test Successful
Check-In items exact quantified	Order Line in Check-In process shows 'X' image for appropriate exact - quantified order line	Order Line in Check-In process shows 'X' image for appropriate exact - quantified order line	Test Successful
Changing Access Rights	System shows access page. Update access process pursued and routed to orders view	System shows access page. Update access process pursued and routed to orders view	Test Successful

Appendix O – White Box Testing Results

Test Case	Tables Affected	Expected Result	Actual Result	Success?
System Login	User	Query of user's username and password in user table where deleted = 0.	Query of user's username and password in user table where deleted = 0.	Test Successful.
View all Companies	Company	Query of company	Query of company	Test Successful.
Create Company	Company	Insert into company table	Insert into company table	Test Successful.
View Company	Company	Query of company table for a particular id	Query of company table for a particular id	Test Successful.
Update Company	Company	Update certain company fields for a particular id value	Update certain company fields for a particular id value	Test Successful.
Delete Company	Company	Update deleted field of company table to 1	Update deleted field of company table to 1	Test Successful.
Automatic Company Deletion	Company, Orders, Orderlines	Query of orders and orderlines for a particular company_id and deleted = 1. Delete from company for a particular id.	Query of orders and orderlines for a particular company_id and deleted = 1. Delete from company for a particular id.	Test Successful.
View all Users	User	Query of users	Query of users	Test Successful.
Create User	User	Insert into user table	Insert into user table	Test Successful.
View User	User	Query of user table for a particular id	Query of user table for a particular id	Test Successful.
Update User	User	Update certain user fields for a particular id value	Update certain user fields for a particular id value	Test Successful.
Delete User	User	Update deleted	Update deleted	Test

		field of user table to 1	field of user table to 1	Successful.
Automatic User Deletion	Company, Orders, Orderlines	Query of orders and orderlines for a particular user_id and deleted = 1. Delete from user for a particular id.	Query of orders and orderlines for a particular user_id and deleted = 1. Delete from user for a particular id.	Test Successful.
View all unconfirmed orders.	Orders, Company	Select orders where supplier_id = company_id and confirmed = null.	Returns orders and appropriate company_id for unconfirmed orders.	Test Successful.
View all confirmed orders.	Orders, Company	Select orders where supplier_id = company_id and confirmed not null.	Returns orders and appropriate company_id for confirmed orders.	Test Successful.
Create Order	Orders, Company	Insert into orders.	Insert into orders.	Test Successful.
Update Order	Orders	Update order set relevant fields	Update order set relevant fields	Test Successful.
Delete Order	Orders, orderlines, ordercheckin	Delete from order, orderlines and ordercheckin for a particular id.	Delete from order, orderlines and ordercheckin for a particular id.	Test Successful.
Finish Order	Orders	Update order set deleted = 1	Update order set deleted = 1	Test Successful.
View Order	Orders, orderdocument	Select from orderdocument for a particular order code.	Select from orderdocument for a particular order code.	Test Successful.
Check-In Order	Ordercheckin, orders, orderline.	Update ordercheckin for the particular orderline and orders id fields	Update ordercheckin for the particular orderline and orders id fields	Test Successful.
Create Order Line	Orders, Company, orderline, user	Insert into orders. Update order total for a particular id.	Insert into orders. Update order total for a particular id.	Test Successful.
Delete Order	Orderlines,	Delete from	Delete from	Test

Line	Orders.	orderlines for a particular id. Update order total.	orderlines for a particular id. Update order total.	Successful.
View Access	role_access, role, access.	Select from role, access, role_access.	Select from role, access, role_access.	Test Successful.
Update Access	Role, access, role_access	Update role_access for particular role_id and access_id	Update role_access for particular role_id and access_id	Test Successful.
Verify Access	role_access, user	select role and access id where user access rights = role id	select role and access id where user access rights = role id	Test Successful.
Search Sales Person	Orders, orderlines, user, orderdocument	Select order, where orderline matches particular user id for date range.	Returns relevant order and document from orderdocument.	Test Successful.
Search Supplier	Orders, orderdocument	Select order for an order id and specified date range of order confirmed.	Returns relevant order and document from orderdocument	Test Successful.
Search Customer	Orders, orderlines, company, orderdocument	Select order for an order id and specified date range of order confirmed and particular customer.	Returns relevant order and document from orderdocument	Test Successful.

Appendix P – User Acceptance Testing

Functionality Test Case	Success?
Data and Information Requirements	
The system must be capable of storing a supplier's name, address, town, county, postcode, telephone and their fax number.	Test Successful.
The system must store an order's id, code, delivery method, the date the order is required, the date the order was closed, any optional details, who confirmed the order and who the order is meant for.	Test Successful.
The system must store an order line's id, item quantity, item code, size, colour, description, unit price, total, corresponding order id, which employee created the order line and who the order line is meant for.	Test Successful.
The system must store a user's username, full name, company reference, telephone extension, access password and an access rights category assignment.	Test Successful.
The system must store a number of actions that the system will allow and which type of user role can perform those actions. The system will therefore also have to store which users are assigned to a particular access role.	Test Successful.
The system must store a list of order documents once an order has been confirmed. This will consist of the persistence of an id, the order number, the document and date of the created document.	Test Successful.
System Features	
Display an order summary page after a successful login. The system should be able to distinguish which orders are confirmed and unconfirmed, placing them in the correct view accordingly.	Test Successful.
Provide a login screen, for which users will be validated entry before proceeding further into the system.	Test Successful.
Users will be assigned to one of either 'Standard', 'Administrator' or 'Temporary User'. Users will be prompted with a message if their role prohibits a particular action from being performed.	Test Successful.
To alert users of orders which are overdue, at a glance.	Test Successful.
Warn before deleting company, user or order data, allowing users to cancel if desired.	Test Successful.
To validate all fields when data is updated / created. This ensures data integrity by appropriately prohibiting null fields.	Test Successful.
To provide efficient search facilities for locating suppliers, orders and customers.	Test Successful.
Allowing one supplier to obtain many orders. However, one order line will be assigned to only one order.	Test Successful.
No changes allowed to confirmed orders, conforming to the security issues mentioned by the Company.	Test Successful.
Allowing order lines to be added and or deleted for a particular order. The order total should update automatically after order-line amendments.	Test Successful.
Access rights should be able to be updated in real time; changes which should take effect without any reload or restart of the system.	Test Successful.
	Test Successful.

Constraints	
Auto-generate and auto-increment the ID number for a particular record within the system, in particular for order and company tables, as these ID numbers are visible within the application.	Test Successful.
Storage of address entries to a maximum of three lines, excluding the Town, County and Postcode fields.	Test Successful.
The system should not allow an order to be confirmed where no order lines yet exist.	Test Successful.
Enforce numeric integrity constraints on fields which must contain numeric data.	Test Successful.
Enforce the entry of a correct date entry, in form of DD/MM/YYYY.	
Storing a supplier, order or order line record if all the required fields are filled in correctly.	Test Successful.
Usability	
Use of uniform colours and text layout throughout the system, maintaining consistency within the application.	Test Successful.
Model reports closely to the appearance and layout of the current ordering system documents.	Test Successful.
Use of contrasting colour and/or text where necessary, to stand out from the normal appearance of the working system, but without intrusive notifications.	Test Successful.
Avoiding use of horizontal / vertical scroll bars, unless used within direct display of application on screen.	Test Successful.
Positioning controls to prevent excessive time taken between mouse movements.	Test Successful.
Show record updates to user in real time, by refreshing information with the updated information	Test Successful.
Allow a user to cancel an action and 'go back' from wherever a user is within the application.	Test Successful.
Security	
Prohibit users without a valid username and password combination from accessing any part of the system, barring a welcome page.	Test Successful.
Restrict limits to the types of information particular parts of the system accept, in an attempt to strengthen hacker vulnerability.	Test Successful.
Only allow administrators to view / change actions for particular role types that a user can be assigned to.	Test Successful.
Preventing anyone other than administrators to remove or edit user's credentials, including an update of an access role type.	Test Successful.
Sending and receiving of all data via a secure channel, whether within the company LAN or over a Wide Area Network (WAN).	Test Successful.
Performance	
The execution of user demands as rapidly and efficiently as possible.	Test Successful.
The system should allow for concurrent execution of database queries, limiting the maximum number of concurrent transactions to 15 users.	Test Successful.

	Test Successful.
Maintenance	
Allowing administrators to undertake simple routine housekeeping tasks, such as database backing-up.	Test Successful.
Catering for other eventual developments in terms of functionality. The system should allow these further developments to be made with no disruption or effect to any already implemented portions of the system.	Test Successful.

Appendix Q – User Evaluation results

Create Order

User	Time in old system (mins)	Time in new system(mins)
Michael (director)	0.33	1.20
Peter (Sales)	0.58	0.52
Karen (sales)	2.05	0.45
Derek (Distribution)	1.05	1.58
David (directors)	1.24	1.55

Task Involved creating an order with a date required, a delivery method and a comment

Create Order Line

User	Time in old system (mins)	Time in new system(mins)
Michael (director)	0.45	0.26
Peter (Sales)	0.40	0.29
Karen (sales)	0.55	0.33
Derek (Distribution)	0.41	0.48
David (directors)	0.36	0.41

Task Involved selecting order to update and adding order line entering relevant data.

Create Updating Company

User	Time in old system (mins)	Time in new system(mins)
Michael (director)	1.26	0.13
Peter (Sales)	1.55	0.26
Karen (sales)	2.55	0.40
Derek (Distribution)	2.01	0.31
David (directors)	3.10	1.06

Task Involved updating an existing company address and county fields

Search for Report

User	Time in old system (mins)	Time in new system(mins)
Michael (director)	5.41	0.16
Peter (Sales)	3.38	0.13
Karen (sales)	2.37	0.21
Derek (Distribution)	4.15	0.18
David (directors)	3.20	0.33

Task Involved searching for an order by sales person and specifying date range

Appendix R – Purchase Order – Implemented System

Appendix S – Letter of Thanks from Laurence Highman & Uniformity